NOTE TO USERS

The original manuscript received by UMI contains pages with indistinct and/or slanted print. Pages were microfilmed as received.

This reproduction is the best copy available

UMI

If it waren't for the rocks, in its bed
the stream would have no song.
Carl Perkins

University of Alberta

DEVELOPMENT OF AN INTEGRATED HYBRID INTELLIGENT SYSTEM FOR POWER MANAGEMENT

by

Sanjay Madan

©

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Doctor of Philosophy**.

Department of Electrical and Computer Engineering

Edmonton, Alberta
Spring 1998

National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-29068-9

Canadä

University of Alberta

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled **Development of an Integrated Hybrid Intelligent System for Power Management** submitted by Sanjay Madan in partial fulfillment of the requirements for the degree of **Doctor of Philosophy**.

KE Bollinger

Prof. K.E. Bollinger

Prof. O.P. Malik

Prof. M. Rao

Prof. N.G. Durdle

Prof. X. Sun

Date: December 3rd, 1997

To
Seema and Sakshi
Parents and Sisters

Abstract

A majority of recent large-scale system breakdowns have been the consequence of instabilities characterized by sudden voltage collapse phenomena. The main reason for this has been the increase in the transient stability limits of power flows, permitting more power to be transferred over longer distances. This has been made possible by improved protection devices, improved voltage regulators, and the use of Static VAr Compensators (SVC). Higher power flows and use of reactive power compensation have led to higher reactive power losses. This has made appropriate control of high voltage problematic in extreme situations, leading to voltage instabilities which have caused large scale blackouts. Closely associated with the problem of voltage stability is the problem of voltage/VAr violations. If the voltage/VAr problem is left unattended or dealt with improperly, it has the potential to develop into a voltage stability problem and disrupt the system.

Existing Energy Management Systems (EMS) are designed to handle functions such as state estimation, security analysis, optimal power flow, etc.. These functions are performed periodically during the normal operating conditions. This scenario changes during a disturbance. For example, during a large disturbance, the EMS can generate thousands of alarms in a matter of minutes, each alarm indicating a specific event such as opening of a circuit breaker. Human operators are left with the burden of interpreting a large number of messages from the EMS, telephones, and other means of communication. The alarm interpretation and fault diagnosis tasks are further complicated by missing and possibly erroneous information. The operators are also expected to restore service to customers as quickly as possible. Automation of these tasks can help improve the capabilities of a utility system in dealing with emergency conditions.

Emergency situations call for quick diagnosis and decision making by the operator. For complex situations such as above (voltage stability and voltage/VAr violations), the operator quickly reaches a cognitive barrier. Artificial Intelligence (AI) techniques, can be

valuable support tool to operators under these conditions.

There are several ways to help operators overcome this human limitation. First, operators need to understand what is happening on the power system and the AI software can give guidance by showing various scenarios that explain the situation consistently. Operators can then check for the most plausible scenarios, some of which may have been overlooked. Further, since power system operations are filled with many fragmented tasks that are done almost routinely; smart software (that is, AI based) can relieve operators from their routine tasks and allow them to devote their time to more important tasks.

This thesis documents the development and testing of an intelligent system which integrates artificial intelligence techniques with analytical techniques. The package is called the **C**omprehensive **V**oltage and **VAr C**ontroller (CVVC) toolbox. Artificial intelligence techniques used in this research are expert systems, neural networks, and data-mining. The developed system makes two types of decisions:

1. Suggests new settings of controllers for remedying voltage/VAr violations. This decision is made by the **V**oltage **VA**r **C**ontroller **M**ethod (VVCM).
2. Suggests where and how much load to shed for voltage instability case or when there are insufficient resources to correct voltage/VAr violations. This decision is made by the **L**oad **S**hedding **M**ethods.

In addition, the CVVC toolbox incorporates the following:

1. A neural network based system, that predicts new power flows over the transmission lines and transformers, once the settings of controllers have been changed or load-shedding has been done.
2. A data mining based system, for inducing a decision tree for classifying the power system operating mode into *Stable*, *Intermediate Stable* or *Unstable* modes.
3. New and improved models to calculate the sensitivities of dependent variables (reactive power of generators and voltages at the load buses) w.r.t. the control variables (tap setting of transformers, generator voltages, and switchable capacitors/reactors).

Acknowledgements

I am thankful to Prof. K.E. Bollinger for his supervision and financial support. His timely advice and urging to begin writing the *first draft* early helped. His involvement in industry while being in academics was a source of inspiration.

I thank Drs. N.G. Durdle and X. Sun for their support as member of my Ph.D. supervisory committee. They provided valuable inputs to this thesis. I also thank Drs. O.P. Malik and M. Rao as members of my examining committee for their evaluation of this thesis.

Thanks are due to the Commonwealth Scholarship Agencies (AUCC, CBIE, and presently ICCS) and NSERC for the financial support.

Dr. W.-Y. Gu and Mr. T. Greenwood-Madsen of Alberta Power Ltd.; Mr. K. Berg of Edmonton Power Ltd.; Drs. P. Herrington, Y. Mansour, and E. Vaahedi of British Hydro gave some valuable inputs to the project. Thanks are due to them. Mr. Kees den Hartigh, system analyst in our department was of great help in providing computing support over multiple platforms and getting expired MATLAB licenses renewed at the earliest. Thanks.

Dr. Stephan Zefeng Ao, Mr. Albert Son, Mr. Rajat Bhargava, Mr. Emanuel Bocancea and Mrs. Romela Bocancea gave good company in the power system and power electronics laboratory. I am thankful to them. Thanks are due to Dr. C.G. Englefield and his family for making our stay in Edmonton wonderful. Thanks are due to Mr. and Mrs. Mahendra Mehta, Mr. Wenzhi Chen and family, Mr. Rorogu Ngeno, Mr. Kaladhar Voruganti, Mr. Pranob Banerjee, Mr. Arun Bhargava, Mr. Ravindra Gudi, and Mr. Srinivas Padmanabhuni for helping now and then in various forms.

I also wish to thank Mr. G.C. Gupta (my school teacher) of Physics and Chemistry who instilled in me the fascination for science, (whether it was Na darting in water or collecting Cl_2 by the water displacement method), and also to Prof. J.S. Gupta (at the Regional

Engineering College, Kurukshetra) who inspired me to look comprehensively on all issues. These teachers not only taught but also inspired.

My grandfather Late Shri Nand Lal Madan encouraged me to continue my studies after the completion of my B.Sc. (Engg.) at the Regional Engineering College, Kurukshetra. At the Indian Institute of Technology, Bombay, my supervisor for the M.Tech program, Dr. S.K. Banerjee exposed to me the wonders of research and I made the Ph.D. as one of my life goals. Shri Sunder Lal Madan, my grand uncle has been a motivating factor for the completion of my Ph.D.. Thanks to all of them. Thanks are also due to parents and sisters for their help, sacrifices and encouragement in reaching this goal.

I am very thankful and grateful to my wife Seema and our daughter Sakshi (who was born in February 1997) for making it possible to spend extra hours working towards the completion of this project. It was a big and supreme sacrifice on their part. A very special thanks to both of you.

Contents

1 INTRODUCTION 1
1.1 Power System States 2
1.2 Voltage/Reactive Power(VAr) Problems 3
1.3 Artificial Intelligence 5
1.4 Motivation 6
1.5 Objectives 7
1.6 Methodology 7
1.7 Organization of Thesis 8

2 LITERATURE REVIEW 9
2.1 Background 9
2.2 Voltage and Reactive Power (VAr) Control 11
2.3 Expert Systems for Voltage/VAr Violation Problem 11
2.3.1 Comments 19
2.4 Linear Programming for Voltage/VAr Violation Problem 20
2.4.1 Comments 22
2.5 Voltage Stability 22
2.5.1 Background 22
2.5.2 Load-Shedding 23
2.6 Expert Systems for Voltage Stability 24
2.6.1 An Intelligent Load Shedding Approach to Enhance The Voltage Stability of a Power System 24
2.6.2 Rotor Heating Criteria 25
2.6.3 Derivation of Rules from Rotor Heating Criteria 26
2.6.4 Comments 27
2.7 Miscellaneous 27
2.7.1 Line Flows 27
2.7.2 Voltage Instability Prediction 27
2.8 Conclusion to Chapter 2 28

3 ENABLING TECHNOLOGIES 30
3.1 Artificial Intelligence 30
3.2 Expert Systems 31
3.2.1 Characteristics of Expert Systems 31
3.2.2 Components of Expert Systems 32
3.2.3 Heuristics 33
3.2.4 Knowledge Engineer 33

3.2.5 Knowledge Representation Models 33
3.2.6 Problem Solving Strategies 36
3.2.7 General Purpose AI Languages 37
3.2.8 Expert System Development Tool (Shell) 37
3.2.9 Applications of Expert Systems 37
3.2.10 Expert System Implementations in EMS 38
3.2.11 Comments 38
3.3 Machine Learning 39
3.3.1 Induction Learning from Examples 40
3.3.2 Data Mining 41
3.3.3 Data Mining and Machine Learning 42
3.3.4 Data Mining Tools 42
3.3.5 Example of Supervised Discovery 43
3.3.6 Example of Unsupervised Discovery 49
3.3.7 Comments 50
3.4 Artificial Neural Networks 51
3.4.1 Artificial Neural Network 51
3.4.2 Neural Network Architectures 53
3.4.3 Comments 56
3.5 Conclusion to Chapter 3 57

4 AI and POWER SYSTEMS **58**
4.1 Power System Problems 58
4.1.1 Causes 59
4.1.2 Steps in Recovery 60
4.1.3 Power System Features 61
4.1.4 Corrective Measures 62
4.2 Power System Scenario 64
4.3 Role of Operator 65
4.4 Role of Artificial Intelligence 65
4.5 Conclusion to Chapter 4 67

5 MODELS and ALGORITHMS **68**
5.1 Methodology 69
5.2 Formulation of VQSM 71
5.3 Expert System Monitor (ESM) 72
5.4 Formulation of Sensitivity Matrix 74
5.4.1 Sensitivity due to Generator Bus Voltage Change 75
5.4.2 Sensitivity due to Capacitor/Reactor switching 77
5.4.3 Sensitivity due to Tap Change in Transformers 78
5.4.4 Sensitivity due to Tap Change in Transformers - New Model 81
5.5 The VVCM Algorithm 84
5.6 Formulation of Linear Programming for Voltage/VAr Violations 88
5.6.1 Mixed Linear Integer Programming 90
5.7 Formulation of Load-Shedding Methods 91
5.7.1 Load-Shedding Method 1 (LSM-SC) 91
5.7.2 Load-Shedding Method 2 (LSM-LP) 93
5.7.3 Load-Shedding Method 3 (LSM-DI) 94

5.7.4 Comments on Load-Shedding Methods 96
5.8 Generation of Database for Neural Network and Decision Tree 97
5.8.1 Generation of New Values of Variables 98
5.9 Training of Neural Network 101
5.10 Inducing a Decision Tree 102
5.11 Conclusion to Chapter 5 104

6 RESULTS **105**
6.1 Voltage VAr Controller Method(VVCM) 105
6.1.1 Summary on VVCM 137
6.2 Load-Shedding 138
6.2.1 Summary on Load-shedding 153
6.3 Neural-Network Based Power Flows 154
6.3.1 Summary for Neural Network Based System 162
6.4 Data Mining Based Induction of Decision Trees 163
6.4.1 Summary on Decision Tree Induction 181
6.5 Miscellaneous 182
6.5.1 Mixed Integer Linear Programming 182
6.5.2 Discrete and Eliminate Method 183
6.6 Conclusion to Chapter 6 184

7 CONCLUSIONS **185**
7.1 General 185
7.2 Summary of Work Done 186
7.2.1 Summary of Main Findings 188
7.3 Contributions 189
7.4 Future Research Directions 192

Bibliography **224**

List of Figures

1.1 Power System Operating States 2

3.1 Induction Decision Tree for Golf Playing 48
3.2 Model of a Neuron 52
3.3 Single-layer Feedforward Networks 53
3.4 Fully Connected Multilayer Feedforward Networks 54
3.5 A General Recurrent Neural Network 55
3.6 A One-dimensional Lattice of 3 Neurons 56

5.1 Structure of CVVC Toolbox 70
5.2 Transformer Model 79
5.3 Modules of VVCM 87
5.4 Two Bus Power System 95
5.5 Structure of CVVC Toolbox 103

6.1 Control Variables (Case VVC1) 111
6.2 Dependent Variables (Case VVC1) 112
6.3 Error between Predicted and Load Flow Values of Dependent Variables (Case VVC1) 113
6.4 Control Variables (Case VVC2) 115
6.5 Dependent Variables (Case VVC2) 116
6.6 Control Variables (Case VVC3) 117
6.7 Dependent Variables (Case VVC3) 118
6.8 Control Variables (Case VVC4) 120
6.9 Dependent Variables (Case VVC4) 121
6.10 Error between Predicted and Load Flow Values of Dependent Variables (Case VVC4) 122
6.11 Control Variables (Case VVC5) 124
6.12 Dependent Variables, Buses 1-15 (Case VVC5) 126
6.13 Dependent Variables, Buses 16-30 (Case VVC5) 126
6.14 Predicted vs Load Flow Values of Dependent Variables (Case 5) 128
6.15 Control Variables (Case VVC6) 130
6.16 Dependent Variables, Buses 1-15 (Case VVC6) 132
6.17 Dependent Variables, Buses 16-30 (Case VVC6) 132
6.18 Control Variables (Case VVC7) 134
6.19 Dependent Variables, Buses 1-15 (Case VVC7) 136
6.20 Dependent Variables, Buses 16-30 (Case VVC7) 136
6.21 Recommended Values for Load Shedding (Case LS1) 141

6.22 Predicted Values of Dependent Variables for Load Shedding (Case LS1) . . 142
6.23 Chosen Values for Load-Shedding (Case LS1) 144
6.24 Error between Predicted and Load Flow Values of Dependent Variables (Case LS1) 145
6.25 Recommended Values for Load Shedding (Case LS2) 146
6.26 Chosen Values for Load-Shedding (Case LS2) 148
6.27 Error between Predicted and Load Flow Values of Dependent Variables (Case LS2) 149
6.28 Chosen Values for Load Shedding (Case LS3) 150
6.29 Error between Predicted and Load Flow Values of Dependent Variables (Case LS3) 152
6.30 Error between Predicted Power Flow Levels and Load Flow Values (Case NN1) 156
6.31 Error between Predicted Power Flow Levels and Load Flow Values (Case NN2) 158
6.32 Error between Predicted Power Flow Levels and Load Flow Values (Case NN3) 161
6.33 Error between Predicted Power Flow Levels and Load Flow Values (Case NN3) 162
6.34 Decision Tree (Case DT1) 165
6.35 Classification by Production Rules (Case DT1) 169

List of Tables

3.1 Data for Golf Playing 46
3.2 Discovery of Kepler's Law 50

6.1 Control Variables (Case VVC1) 109
6.2 Dependent Variables (Case VVC1) 111
6.3 Predicted vs Load Flow Values of Dependent Variables (Case VVC1) 113
6.4 Control Variables (Case VVC2) 114
6.5 Dependent Variables (Case VVC2) 115
6.6 Control Variables (Case VVC3) 116
6.7 Dependent Variables (Case VVC3) 117
6.8 Control Variables (Case VVC4) 119
6.9 Dependent Variables (Case VVC4) 120
6.10 Predicted vs Load Flow Values of Dependent Variables (Case VVC4) 121
6.11 Control Variables (Case VVC5) 123
6.12 Dependent Variables (Case VVC5) 125
6.13 Predicted vs Load Flow Values of Dependent Variables (Case VVC5) 127
6.14 Control Variables (Case VVC6) 129
6.15 Dependent Variables (Case VVC6) 131
6.16 Control Variables (Case VVC7) 133
6.17 Dependent Variables (Case VVC7) 135
6.18 Recommended Values for Load Shedding (Case LS1) 140
6.19 Predicted Values for Load Shedding (Case LS1) 141
6.20 Chosen Values for Load-Shedding (Case LS1) 143
6.21 Predicted vs Load Flow Values of Dependent Variables (Case LS1) 144
6.22 Recommended Values for Load Shedding (Case LS2) 146
6.23 Chosen Values for Load-Shedding (Case LS2) 147
6.24 Predicted vs Load Flow Values of Dependent Variables (Case LS2) 148
6.25 Chosen Values for Load Shedding (Case LS3) 150
6.26 Predicted vs Load Flow Values of Dependent Variables (Case LS3) 151
6.27 Predicted vs Power Flow Levels (Case NN1) 155
6.28 Predicted vs Power Flow Levels (Case NN2) 157
6.29 Predicted vs Power Flow Levels (Case NN3) 160
6.30 Summary of Data Mining Results 182

List of Symbols & Abbreviations

b	bias
k	transformer tap ratio setting
m	number of load buses
n	number of generator buses
t	number of transformers
A	Dependent Variables
ALS	Predicted Values of Dependent Variables after Load-Shedding
$CVVC$	Comprehensive Voltage VAr Controller Toolbox
DM	Data Mining Based System
Δ I	Delta Current
ILP	Integer Linear Programming
LP	Linear Programming
$LP1$	Linear Programming - Method 1
$LP2$	Linear Programming - Method 2
LSM	Load Shedding Method
$LSM(DI)$	Load Shedding Method Based on Delta Current Injections
$LSM(LP)$	Load Shedding Method Based on Linear Programming
$LSM(LP1)$	Load Shedding Method Based on Linear Programming Method 1
$LSM(LP2)$	Load Shedding Method Based on Linear Programming Method 2
$LSM(SC)$	Load Shedding Method Based on Sensitivity Coefficients
$LSM(SC-H)$	Load Shedding Method Based on Sensitivity Coefficients with Heuristics
$MILP$	Mixed Integer Linear Programming
NN	Neural Network Based System
$NSTEPS(R)$	Number of Step Changes (rounded) Required for Remedying Voltage/VAr Violations
P_g	Real Power Generation
P_l	Real Power Load
Q_g	Reactive Power Generation
Q_l	Reactive Power Load
S	Sensitivity Matrix
V_g	Generator Bus Voltage
V_l	Load Bus Voltage
Sum_LS	Sum of Reactive Power Load-Shed
$VVCM$	Voltage VAr Controller Method
$VQSM$	Voltage reactive power (Q) Sensitivity Method
W	Synaptic Weight
X	Control Variables
XLS	Recommended Value of Load-Shedding
Y	Admittance Matrix

Chapter 1

INTRODUCTION

In its simplest form, a power system consists of generators, loads, and transmission lines, which transmit power from the generators to the loads. The transmission lines form a network with generators and loads at different points. The flow of power through the transmission lines is regulated by the action of various controllers (e.g. tap-changing transformers, switching capacitors and reactors, generator-voltage control, turbine-governor control, etc.). A variety of constraints govern the actions of these controllers, one of which is the thermal limit of power flow on the transmission lines.

When a disturbance occurs, manual or automatic control actions are applied to return the system back to the *normal* or *secure* state as soon as possible. A *disturbance* refers to any system condition which requires a transmission system component or element to be taken out of service immediately, either automatically or as soon as switching can be performed. A disturbance also refers to abnormal transmission system behavior being characterized by overloads, abnormal under- or over- voltages or frequency, a significant loss of area load, loss of supply continuity to load buses/sub-stations, loss of generation or loss of area stability. A clear understanding of system conditions during power system disturbances is needed to devise effective corrective measures. One needs to define the operating constraints and classify the power system operating status, into distinct classes. This enables one to analyze the operation of the power system during a disturbance (diagnosis), and to select actions (remedies) from that analysis to restore the system to a normal, healthy state. Power system operation is subject to *equality* and *inequality* constraints. The requirement to keep the loads and generation of both the active and the reactive power components balanced are examples of equality constraints. The requirements to keep the system parameters such as voltages, line flows, and bus angles, within the specified limits are referred to as inequality constraints.

1.1 Power System States

Power system operating conditions may be classified as having one of the five states as shown in Figure 1.1.

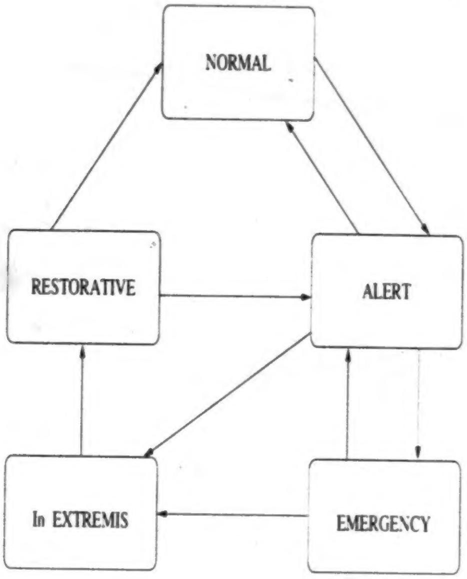


Figure 1.1: Power System Operating States

In the *normal* state, all system variables are within a specified range and no equipment is overloaded. The system enters the *alert* state if the security level falls below a certain level of adequacy, or if the possibility of a disturbance increases (e.g. adverse weather conditions such as an approaching severe storm). In the alert state all system variables are within the normal range and all constraints are satisfied. In this state the system has been weakened to a level where a disturbance may cause an overloading of equipment that may put the system into an *emergency* state. The system may enter the *in-extremis* (extreme emergency) state if the disturbance is very severe. In the emergency state, voltages at many buses are low and/or equipment loadings exceed the short-term ratings. The system is still intact and may be restored to the alert state by initiating emergency control action (e.g. fault clearing, excitation control, load-shedding, etc.). The system may go into an in-extremis state if the above measures are not applied or are ineffective. In this state the system loses its integrity; that is, it breaks down into two or more islands or fails to service the load. In this state, control actions such as load-shedding and controlled system separation are actuated to save as much of the system as possible from widespread blackout. In the *restorative* state, control actions are taken to restore all the facilities and to restore the system load. The system transits from this state to the alert or the normal state depending upon the initial conditions [Kundur94].

[Zaborszky79] further sub-divides the *emergency state* by identifying three crises that may occur during an emergency scenario. These three, the *stability*, the *viability*, and the *integrity crisis*, define the type of control action required by characterizing the time frame and the general type of action required. The *viability crisis* is an imbalance in generation, consumption, and transmission capabilities, resulting in constraint violations that cannot be tolerated indefinitely and which lasts from a few seconds to many minutes from the crisis onset. The *integrity crisis* includes the *in-extremis* and *restorative* states where power system integrity has been lost. The *stability crisis* is a transitional emergency condition in which the system integrity is at risk, lasting for a few seconds from the crisis onset. Stabilizing controls maintain synchronism by fast action, within fractions of a second, generally by using local control loops with minimal computation and communication requirements.

1.2 Voltage/Reactive Power(VAr) Problems

Voltage instability is one problem which causes the system to be placed in an emergency or in-extremis state. Voltage stability is the ability of the power system to maintain steady-state acceptable voltages at all buses in the system under normal conditions and after being subjected to a range of disturbances. A system is in a state of *voltage instability* when a disturbance, such as an increase in load demand or a change in system condition, causes a progressive and uncontrollable drop in voltage. The main factor causing voltage instability is the inability of the power system to meet the demand for reactive power. A power system at a given operating state, and subject to given disturbances, undergoes *voltage collapse* if post-disturbance equilibrium voltages are below acceptable limits [Kundur94, Taylor94].

It is alarming to note that mature systems are likely to suffer from voltage stability problems. This problem arises because of the following reasons: Present day utilities face conflicting demands. On the one hand because of economic and environmental reasons there are problems in adding to the generation side, or in getting *right of way* for more transmission lines. On the other hand the demand for secure and reliable power to customers keeps on increasing. This leads to the interconnection of many power generating systems, making the overall power system more complex. Additionally, power equipment is often operating at or near maximum rated capacity. Under these operating conditions any power system disturbance has the potential to threaten the integrity of the system. It is important that under these distress conditions proper action is taken by the operating personnel. On the contrary, the operators reach a cognitive barrier because of stress and due to lack of concise and correct information about the system. The Voltage and VAr violation problem is closely related to voltage stability. This problem arises when specified voltage limits at the load buses and/or the specified VAr limits of the generators are violated with the system still being voltage stable. However, if voltage/VAr violations are left unattended or dealt with improperly, they may develop into voltage stability problems, and disrupt the system. One prime aim of this thesis is to deal with problems of voltage stability and voltage/VAr violations in a comprehensive manner.

The voltage stability problem is classified in the *in-extremis* class. The voltage/VAr violation problem may be put in the *emergency* class or in the *alert* class depending upon the severity of the violation(s).

When the system is in the *in-extremis* state, certain control actions are performed which attempt to restore the system back to the normal state. Load-shedding has been recognized as one method of regaining the normal stable state. Load-shedding is however, the last resort to save the system from losing synchronism. It has been advocated as the last minute solution to prevent voltage collapse. Load-shedding may be defined as a set of controls to save the system from further deteriorating into an in-extremis state from either the emergency or the normal state. The decision-making process as to how much load to shed, and where, varies from situation to situation and does not follow any algorithmic pattern. It is very heuristic in nature and involves a considerable amount of engineering judgment. Very often, it is dependent upon the particular expertise of the operator handling the crisis. It may be remarked that while load-shedding has been used effectively for decaying frequency cases, its use for under-voltage is still in its infancy [Taylor94].

In general, voltage/VAr violation problems are corrected by selecting an appropriate choice and setting of the controllers. This process of selection is characterized by making a search for the *best controller*. The best controller is one which provides maximum relief to the system having voltage/VAr violation problems. Since voltage/VAr violations signals the onset of a lowering of voltage security of the system, any effort to remove them enhances voltage stability [Madan96][1].

This research work investigates ways of imparting assistance to operators when a power system is under distress, so that they can make beneficial decisions on voltage and VAr control measures, and use load-shedding only as a last resort. When required, however, assistance is also provided for load-shedding decisions, e.g. when, how much, and where, to shed the load. This assistance or decision support will be provided using artificial intelligence techniques in conjunction with analytical techniques.

1.3 Artificial Intelligence

Artificial Intelligence (AI) is that part of computer science concerned with the design of intelligent computer systems, that is, systems that exhibit the characteristics that we associate with intelligence in human behavior - understanding language, learning, reasoning, solving problems, and so on [Jackson90].

Artificial intelligence is concerned with programming computers to perform tasks that are presently done better by humans, because they involve high-level mental processes such as perceptual learning, memory organization, and judgmental reasoning. Therefore, writing a program to perform complicated statistical calculations would not be seen as an artificial intelligence activity, while writing a program to test hypotheses would. Most people are not very good at doing long calculations by hand, whereas computers excel at such tasks. On the other hand, devising good experiments to test hypotheses is a skill that the research scientist derives partly from training and partly from experience. Programming a computer to perform such a task is non-trivial. Artificial intelligence techniques can be a valuable support tool to operators under conditions when their comprehension limits are reached. Artificial intelligence encompasses Expert Systems (ESs), Artificial Neural Networks (ANNs), and Data Mining (DM). These sub-fields of artificial intelligence have been used in this research.

An *expert system* is a computer program that assimilates knowledge obtained from some expert(s) and reasons with this knowledge with a view to solving problem(s) and/or giving advice [Rich91]. Thus, expert systems are software packages which translate human expertise into computer programs. Portability of software makes the use of expert systems very attractive where human expertise is scarce or costly or is likely to be lost through mobility.

---

[1] "Artificial Intelligence Aided Voltage Stability Enhancement," *Canadian Conference on Electrical and Computer Engineering*, May 26th - May 29th, 1996, Calgary, Alberta, pp. 392-395.

An *artificial neural network* is a massively parallel distributed processor that has a natural propensity for storing experimental knowledge and making it available for use when required [Haykin94]. It resembles the brain in two respects:

1. Knowledge is acquired by the network through a learning process.
2. Interneuron connection strengths known as *synaptic weights* are used to store the knowledge.

Data mining is the search for *new* or *interesting* facts, relationships, and rules from large archives of data. The idea is to find a pattern *hidden* among the mass of data [Frawley92]. This is analogous to extracting mineral(s) from the ore. Data mining is a powerful means to extract useful rules for building expert systems.

1.4 Motivation

In recent years, voltage stability has emerged as an important consideration in the planning and operating of power systems world-wide. Economic and environmental constraints are forcing systems to be operated under increasingly stressed conditions which make systems susceptible to voltage instability. Further, the increased complexity of modern day interconnected power systems, places a heavy stress on the operators managing these power systems, making it difficult to achieve an optimal performance for the overall system. This means that the utilities are operating under tighter margins and closer to the limits as time progresses. This is only expected to become more critical in future years. These developments will contribute towards the cognitive barrier of the operator.

Therefore, if some pre-processing is done (using AI), to present the operators with a diagnosis of the problem and a list of suggestions, they would be in a better position to make a decision based on an objective analysis. An intelligent system, installed in an Energy Management System/ Dispatcher's Training Simulator (EMS/DTS) environment, can provide suitable assistance when voltage/VAr violation and/or voltage stability problems occur.

This system can be designed to adapt to newer situations. The proposed system can perform the necessary pre-processing and acts as a reliable EMS operator's assistant.

1.5 Objectives

The primary objective of this dissertation is to propose, justify, and describe the development of a system that takes an intelligent approach to solving the voltage/VAr violation and voltage stability problem. The following steps are taken to realize the research objectives:

1. To understand and simulate the behavior of a power system under various operating conditions, especially under emergency conditions, and to identify corresponding corrective actions.

2. To perform a critical review of existing solution techniques and how they are presently implemented.

3. To develop new and intelligent techniques after examining how existing techniques can be improved.

4. To understand and acquire operational and heuristic knowledge by interviewing and by corresponding with relevant personnel (operators and experts) in power utilities, in order to build a knowledge base for the expert system.

5. To develop a system called the **C**omprehensive **V**oltage and **V**Ar **C**ontroller (CVVC) toolbox which will help the operator in decision making when the power system has a *voltage stability* or *voltage/VAr violation* problem.

6. To investigate the possibility of making the CVVC more robust and more useful by incorporating appropriate artificial intelligence techniques, such as neural networks and data mining, in conjunction with an expert system.

1.6 Methodology

The following steps have been identified, to achieve the objectives of this research [Madan97d][2].

1. **Determine the Nature of the Problem:** A load flow program is run. Based on the load flow results it is determined if the system has any *voltage stability* or *voltage/VAr violation* problems. If these problems are not there, then the program reports the status of the power system as normal. The program continues to monitor the power system. If the power system exhibits a problem of *voltage/VAr violation* or *voltage stability*, then corrective actions are invoked.

---

[2] "Development of an Integrated Intelligent System," *IEEE PACRIM'97*, Victoria, B.C., August 1997, pp. 827-830.

2. **Actions for voltage/VAr violation problem:** If the system is *voltage stable* but has a *voltage/VAr violation* problem, the problem may be solved by selecting a proper setting of the controllers to improve the voltage and reactive power profile. The controllers are:

    (a) Synchronous Condensers or Capacitors/Reactors

    (b) Generators

    (c) Tap Changing Transformers.

    If the above resources are unable to resolve the voltage/VAr violation problem satisfactorily, load-shedding may be recommended, depending upon the severity of the violations.

3. **Actions for voltage stability problem:** If the system is *voltage unstable*, one may have to resort to *load-shedding*. Load-shedding may be defined as a set of controls to save the system from further deteriorating into an *in-extremis state* (extreme emergency) from either the *emergency* or the *alert state*.

Following these steps the embedded neural network predicts new power flows over the transmission lines and transformers.

1.7 Organization of Thesis

The thesis is organized as follows: This introductory chapter has introduced the problems of voltage/VAr violation and voltage stability. A power system classification scheme was discussed and the problems under study were classified. Further, the artificial intelligence techniques that will be used in solving the problems are also outlined. The next chapter reviews the relevant literature. This chapter also examines possible improvements that can be made to existing power system models and algorithms. Chapter 3 describes the enabling technologies which will be used to realize the objectives of this research. Chapter 4 presents the emerging power system scenario. This chapter explains how various artificial intelligence techniques are used in providing assistance to operators during critical situations. Chapter 5 gives details of the modeling of power systems and the programs developed for dealing with voltage/VAr violation and voltage stability problems. This chapter also outlines the contributions made by the author, by way of new or improved models and algorithms for the problems under consideration. Chapter 6 presents the results of the modeling presented in Chapter 5. The last chapter reports conclusions and identifies future research directions.

Chapter 2

LITERATURE REVIEW

2.1 Background

Power system stability is defined as the property of a power system that enables it to remain in a state of operating equilibrium under normal operating conditions and to regain an acceptable state of equilibrium after being subjected to a disturbance.

Depending upon the system configuration and operating mode, instability of the power system may be manifested in many different ways. Traditionally, the stability problem has been one of maintaining synchronous operation. Power systems rely on synchronous machines for generation of electric power. Therefore, a necessary condition for satisfactory system operation is that all synchronous machines remain in synchronism *or in step*. This type of stability is influenced by the dynamics of generator rotor angles and power-angle relationships.

Instability may also be encountered without loss of synchronism. For example, a system consisting of a synchronous generator feeding an induction motor load through a transmission line can become unstable because of collapse of load voltage. Maintaining synchronism is not an issue in this instance; instead the concern is stability and control of voltage. Voltage instability can also occur in loads covering an extensive area and supplied by a large system.

A criterion for voltage stability is that, at a given operating condition for every bus in the system, the bus voltage magnitude increases as the reactive power injection at the same bus is increased. A system is voltage unstable if, for at least one bus in the system, the bus voltage magnitude decreases as the reactive power injection at the same bus is increased.

Put in other words, a system is voltage stable if the V-Q sensitivity is positive for every bus and voltage unstable if the V-Q sensitivity is negative for at least one bus.

Voltage stability problems normally occur in systems experiencing heavy electrical loads. The disturbance leading to voltage collapse may be initiated by a variety of causes. The main factors contributing to voltage collapse are:

- the strength of the transmission networks,
- power transfer levels,
- generator reactive power/voltage control limits,
- load levels,
- characteristics of reactive compensation devices, and
- the action of voltage control devices such as tap changing transformer [Kundur94, Taylor94].

The underlying problem is inherent weakness in the power system to transmit reactive power over long distances.

As stated earlier, voltage/VAr violation problems (with the system being voltage stable), if left unattended, may develop into voltage stability problems. This thesis addresses the development of support tools to assist operators in remedying *voltage/VAr* violation and *voltage stability* problems, in a comprehensive manner. To a large extent, the problems of voltage/VAr violations and voltage stability have been treated separately, in the literature. One reason for this is the recent prominence of voltage stability problems.

This chapter has the following objectives:

1. Review existing literature describing applications of artificial intelligence and mathematical techniques (e.g. linear programming) for voltage/VAr violation problems.
2. Review applications of artificial intelligence for voltage stability problems.
3. Review other relevant topics, such as methods to predict voltage stability.

2.2 Voltage and Reactive Power (VAr) Control

The voltage/VAr control has the following objectives for efficient and reliable operation of power systems [Kundur94]:

1. Utility and customer equipment are designed to operate at a certain voltage rating. Prolonged operation of the equipment at voltages outside the allowable range could adversely affect their performance and possibly cause them damage, and/or reduce their life. Hence voltages at the terminals of all equipment in the system should be within acceptable limits.
2. System stability is enhanced to maximize utilization of the transmission system.
3. The reactive power flow is minimized so as to keep IR^2 and IX^2 losses to a minimum. This allows efficient use of the transmission facilities, mainly for the transmission of active (real) power transfer.

2.3 Expert Systems for Voltage/VAr Violation Problem

This section reviews methods which have used artificial intelligence techniques (primarily expert systems) for the voltage/VAr violation problem, and thereby enhanced voltage stability. It is attractive to apply expert systems technology to voltage/VAr violation problem for the following reasons [WorkingGroup88].

1. The presence of a variety of control devices (such as tap-changing transformers, generators, synchronous condensers/capacitors, reactors) to control deviations in the system means that a search can be made to choose a good controller for a given situation. Typically, artificial intelligence problem solving techniques are search oriented and provide ways to prune the search space by using heuristics.
2. Expert systems allow human expertise to be encoded in their knowledge bases.
3. If voltage/VAr violation problem is tackled by analytical methods alone, this necessitates a large amount of computation, using repeated load flows or the study of V-Q and P-Q curves.

Few expert systems have been reported in the literature whose sole objective is to solve the voltage/VAr violation problem and reviewed in [Madan96][1]. These expert systems work in conjunction with numerical methods and are based on sensitivity calculations. The basic algorithm is given below. The variations in these systems are discussed after the algorithm.

---

[1] "Artificial Intelligence Aided Voltage Stability Enhancement," *Canadian Conference on Electrical and Computer Engineering*, May 26th - May 29th, 1996, Calgary, Alberta, pp. 392-395.

Voltage/VAr Violation Correction Algorithm

The knowledge base is formed from load flow results. Limits on system variables (for example, the reactive power of the generator buses, location of reactors and capacitors at the load buses, and tap setting of tap changing transformers), load bus voltages, and sensitivity tables are also stored. The mathematics involved in formulating the sensitivity matrix is given in Chapter 5.

The steps in the algorithm are:

1. *Identify buses with voltage violations. The bus with the maximum deviation is identified as the target bus.*

2. *For the target bus, choose a controller which has maximum product of sensitivity and the available control margin. The control step is so selected that no further violations occur at other buses as a result of this control action. The effect of this change in controller settings on voltage of load buses and reactive power of generator buses is computed.*

3. *Repeat this process till all voltage violations are eliminated.*

In the literature, few systems have been developed using the above algorithm using artificial intelligence languages, primarily *Prolog*. Prolog is a procedure-oriented language. Therefore, these implementations search for the solution using production rules. The important production rules are summarized below:

Production Rules for Voltage and VAr Control

1. **IF** there are no voltage violations
   THEN exit.

2. **IF** system voltage violations are present,
   THEN identify the bus with the largest violation as the target bus.

3. **IF** the target bus has been identified,
   THEN select the control action of the controller which has the highest value for the

product of the sensitivity of control actions and the available control margin of the controllers.

4. **IF** a control action is selected,
   THEN set up the desired voltage increment for the target bus,
   AND round the control increment to the nearest available physical step.
   AND Update the load bus voltages and reactive power generation of generator buses.

5. **IF** the control strategy improves the network voltage and the reactive power generation at the generator buses are within the operational limits,
   THEN accept the selected control action and store the latest position of the controller,
   AND update the control margin of the used controller.

6. **IF** the latest control margin for any controller is zero,
   THEN remove that corresponding controller from the list of the available controllers.

7. **IF** the updated controller position introduces new violations,
   THEN reduce the control increment until the new violations are eliminated,
   AND update the voltage and reactive generation using the new control increment.

8. **IF** the reactive power generation of generators are beyond the operational limits,
   THEN reduce the control increment until the limit is enforced.
   AND update the system voltage and reactive generation using the new control increments.

9. **IF** reactive power generation of a generator bus has reached its limit,
   THEN remove this generator bus from the list of control actions.

10. **IF** the target bus is within the limits,
    THEN re-evaluate the system violations, designate a new target bus, and repeat this process.

Discussion

[Belhadj96] presents an integrated intelligent power system controller. The developed system has two major parts: The first part consists of application programs such as load flow, bus sensitivity, and steady state voltage stability index program. The second part consists of a knowledge-base program. An expert system shell called CLIPS (an expert system shell) [Clips93] (*C Language Integrated Production System*) is used to construct the knowledge base. Some more details about the CLIPS is given in Appendix 'A'). The developed system has three objectives: to regulate voltage, alleviate transmission line overload, and make corrections for steady state voltage instability. The basis for correction is the use of analytical techniques and sensitivities of the power system variables to existing VAr control tools. Results have been presented for 6 and 118 bus systems (2 violations only).

[Cheng88] presents an expert system for voltage and VAr control and is based on the use of sensitivity trees. For each load bus there is a sensitivity tree. Each tree represents the entire relationship between a particular bus voltage and all the control measures and secondary variations of the other bus voltages. The root of the tree is the controlled bus voltage. Each root connects to the control measures through branches. The sensitivity factor associated with the controlled bus voltage and the control measure is given in the corresponding branch. The second level of the tree gives the effect of the control measures on all other load bus voltages. In the second level, each control measure is connected to all other load bus voltages by sub-branches. The sensitivity factors between the control measures and the bus voltages are shown as sub-branches. Similar trees can be made for the reactive power dispatch control. In these trees, the reactive powers drawn at each load bus become the control measures. The roots and the second level leaves are the same as the voltage control sensitivity trees. Appropriate sensitivity factors corresponding to the reactive power dispatch and the bus voltage are used.

In [Cheng89] the above expert system was re-implemented, this time making use of a Data Base Management System (DBMS). This expert system shares the database of the DBMS as one of the application routines through an interface and hence makes use of the database directly. This reduces the required memory space and the probability of error occurrence. The system has been implemented in Quintus Prolog.

[Chowdhury93] presents an expert system to remedy voltage and branch overload problems. The developed expert system is combined with economic dispatch to provide real-time security. The strategy consists of running an economic dispatch which identifies the constraint violations in bus voltage magnitudes and in line flows, and generates a set of optimal generations. The expert system then determines the best possible control measures using rules for voltage and line flow control. The goal of the expert system is to expeditiously remove voltage and branch overload problems. Sensitivities are calculated based on repeated load flows and averaging them. Results have been presented for a modified IEEE 24 bus system.

[Godart91] describes two rule-based expert systems which predict the impact of voltage control devices on bus voltages using a *reactive path concept*. The reactive power path is the one in which the major reactive power flow will occur, following any compensator action. The first technique outlined in the paper determines a set of buses on which a particular controller has a significant impact. The second technique determines the two reactive power compensators which are best suited to alleviate voltage violations at a particular bus. The two techniques described in [Godart91] are implemented as tools for the **V**oltage **C**ontrol **E**xpert **S**ystem, (VOCES) and use OPS/83 [Forgy83].

The *reactive path concept* in [Godart91] can be explained as follows:

Across a line of complex series impedance, Z = R + jX, the relative variation of the voltage magnitude due to complex power flow, S = P + jQ, is approximated by:

$$\Delta V/V = (RP + XQ)/V^2 = XQ/V^2 \qquad (2.1)$$

where $X/R \geq 1$ and V is the voltage magnitude at the origin of the line. Therefore, in transmission systems, bus voltage magnitudes are strongly coupled with reactive power flows. Control parameter changes at the reactive power compensators modify reactive power flow in the power system. Hence, from the above equation, the greater (positive or negative) the reactive power flow from a controller to a particular load is, the greater is the impact from this controller on the load voltage.

Due to reactive power losses in transmission lines, the effect of any reactive power compensation is limited to a region surrounding the particular reactive power device. This region is called Reactive Power Control (RPC) region. The purpose of a RPC region search is to associate with every controller a set of buses such that following any control action by a particular controller:

- The voltages within the region are significantly affected.
- No voltages at buses outside this region are significantly affected.

The compensators' impact is limited to its RPC region. This reduces the number of conflicts to be resolved. RPC regions are found using a heuristically guided search. Among all reactive power flows leaving the control bus, a fraction of the maximal one, in absolute values, $\alpha\|Q\|$, is retained as a threshold to determine the level of reactive power sufficiency. The value of α has to be fine tuned by an expert for each particular power system.

[Goes92] presents two expert systems designed to serve as a control center operator's assistant. The first system helps operators in solving overload problems and the second assists in solving voltage problems. The proposed expert systems make use of *distribution factors*, which are a form of *sensitivity factors*. The authors state that the use of distribution factors circumvents problems introduced by the linearization of distribution factors, but this does not seem to be correct. The system has been implemented in Prolog and tested on a 634 bus model of an actual electrical system. The given example is for 10 voltage violations and does not include the action of tap-changing transformers, or the effect on reactive power of generators.

[Hsu92] describes an expert system that combines the operator's dispatching rules and the integer linear programming routine (ILP) for voltage control of the Taiwan Power Company. The authors contend that the sensitivity approach is very efficient and effective when the number of violations is small (violations for upto five or fewer buses). However, if there are more violations the sensitivity approach can suffer from the difficulty that new violations can be introduced when a corrective action is taken. The ILP is formulated to cater to this problem and to minimize the number of switching of capacitors/reactors and/or adjustments to transformer taps.

The program in [Hsu92] has the following limitations: Only switchable capacitors/reactors and adjustable transformer taps are used as control variables. In addition, only the effect of control variables on load bus voltages is evaluated. The ILP approach does not consider generator VAr limitations. Transformer taps are given the least priority as frequent use of transformer taps reduces the life expectancy of the transformer taps. Further, the effect of transformer tap adjustment is widespread. The end product is an expert system which uses heuristic algorithms for simple cases and resorts to ILP routine for complicated cases. Heuristics rules have been incorporated in formulating the ILP problem.

[Keyhani89] presents a rule-based approach for decentralized voltage control by constructing a *local network model* with respect to a target bus (bus needing voltage correction). A local network consists of a group of *internal* buses, *boundary* buses and *connecting* lines. The load (P-Q) and inactive generator (P-V) buses are the internal buses. Active P-V buses and the slack bus are the boundary buses. In effect a local network is a portion of the bulk power system that has its boundary cut at the active buses. A voltage violation within the local network does not affect any facilities outside the network and a boundary bus control action only affects the adjacent local networks. The paper reports results for a 118 bus test system.

[Liu90] outlines the development of a **V**oltage **C**ontrol **E**xpert **S**ystem (VCES). The VCES is based upon sensitivity factors. VCES has been tested on the IEEE 30 bus system and this approach has been compared with a linear programming approach. VCES performs better than the linear programming for severe voltage violations. VCES was extended to have a learning module built into it by using induction learning from examples [Wang90]. The research in [Wang90] has two objectives: The first application is to obtain a decision tree which identifies a proper value of the low voltage limit. This is useful because estimation of the control amount is based on sensitivities of voltages with respect to controls. The sensitivities are inaccurate when voltages are low in a power system. The second application is an attempt to use inductive learning to identify critical operating conditions which may cause voltage problems in a power system.

VCES has been further extended as a Research and Development Project and [Barruncho92] describes an implementation and operational experience in a real time environment. The new system incorporates a powerful, general purpose OPS83 (an expert system shell) [Forgy83], workstation and a multi-objective **R**eactive **OP**timization **F**unction (ROPF). ROPF is based on a numerically robust MINOS package. The new system has been termed **R**eactive **M**anagement **V**oltage **C**ontrol (RMVC) whose knowledge base part is based on the VCES. The RMVC runs after each valid state estimation solution, typically on a 10 minute basis or is event driven by topological changes.

[Ramos95] presents an integral tool to assist the operator in reactive power dispatching for voltage profile improvement and reactive power loss minimization. The method deploys a strategy of shifting several control variables in proportion to their *energy coefficients* (ECs), which favors the use of devices with higher sensitivities. The algorithm tries to choose a strategy which causes control variables to return or keep close to their normal values. This tool has been developed based on the MINOS package and the ART expert system shell, which makes it easy to integrate many computationally-intensive routines written in *C*.

Most control actions are conditioned by the tendency of loads to increase, decrease or remain steady. This seems logical; for example, the power plant operator will not be willing to change the voltage setpoint if he/she takes it for granted that the opposite action will be necessary in, say, twenty minutes. To this end, there are five representative 24-hour load patterns (Friday, Saturday, Sunday, Monday, and other week days).

[Singh93] presents an expert decision support system for voltage control in a power system network. The control strategy is *centralized* and has been implemented in FORTRAN and tested for two loading conditions of the IEEE 14 and 30 bus systems. A comparison is made with results using a *decentralized* approach [Keyhani89]. The authors claim that the results obtained by this method are more accurate and require less control measures to correct the voltage compared with the decentralized method.

[Yokoyama93] presents an approach for voltage and VAr control in a power system for enhancing voltage security using the theory of fuzzy sets in conjunction with an expert system. In conventional expert systems one needs perfect matching of input(s) to predetermined conditions. On the other hand in the proposed methodology, control heuristics are expressed by IF THEN rules, where the premises are given by linguistic variables, measuring the proximity of a given input to a certain condition to be satisfied, as a grade of membership in fuzzy sets. Every premise included in a rule has a numerical value, rather than a true or false state as in a conventional rule-based system. This enables relaxed pattern matching and flexible application of numerical models. The authors contend that this methodology is only suitable for bus voltage deviations that arise under stable operating conditions. Results have been presented for the IEEE 14 bus system.

[Zalapa95] proposes a methodology for steady state conditions to determine the output or setting of existing VAr/voltage control devices so that the allocation of reactive reserves guarantees that the system does not move towards voltage collapse as demand changes and that it will cope under credible contingency conditions. The method involves defining *reactive areas* by identifying *weak* circuits and then to assess the distance to voltage collapse by *Q-distances* method. The Q-distances stand for various indices to low voltage (LV) and voltage collapse (VC) conditions, which are computed using the Q-V curves. For example, Q-distance to VC implies maximum increase of reactive demand at the node that the system can withstand before it becomes voltage unstable. An outage screening algorithm is run for VAr rescheduling, enhancing security and limiting any violations in a robust manner. This proposed global strategy had been prototyped on a computer by integrating algorithmic software and symbolic programming. According to the authors, although the methodology is basically algorithmic, the use of knowledge-based system technologies made prototyping simpler; *it is not clear why this should be so.*

2.3.1 Comments

The basic algorithm in [Belhadj96, Cheng88, Chowdhury93, Godart91, Hsu92, Keyhani89, Liu90, Ramos95, Singh93, Yokoyama93, Zalapa95] for voltage and VAr regulation described in section 2.3. has some limitations which are listed below. Proposals to overcome these limitations and added features developed as part of this thesis are also discussed. This basic algorithm has been modified, improved and extended to form the Voltage VAr Controller Method (VVCM).

- The basic algorithm cited above has the possibility of falling into an infinite loop because a control action at a bus may potentially violate other bus limits. Hence, using this type of expert system in a stand alone mode is risky. It may not find any solution. *To counter this limitation, in this thesis, use of linear programming and mixed integer linear programming modules has been made. These modules cater to severe violations or when the above mentioned algorithm is likely to fall into an infinite loop.* [Liu90] also uses the same feature.

- The model is based on sensitivity factors which give only linear approximations to the increments of reactive power of generators and voltage of load buses. This is not valid when increments are large. Sensitivity factors giving exact increments can only be achieved through load flow analysis, which is time consuming. *To counter this limitation, in this thesis, the developed system (VVCM) has a facility to recognize that the current power system operating state is much different from the base case, and sensitivities can be recalculated.*

- Sensitivity factors calculations ignore the resistance of transmission lines. Hence, if X/R of transmission lines is low, the sensitivity factors are more erroneous. *To counter this limitation, in this thesis, the sensitivities are calculated using the complex* *impedance of the system, rather than only the reactance of the system. An improvement was noted.*

- The developed VCCM in this thesis, has been written primarily in MATLAB and CLIPS[2]. In the literature, similar implementations (reviewed in [Madan96][3]) have been developed using AI languages, primarily Prolog, which is a logic oriented language. [Singh93, Yokoyama93] implementations are in FORTRAN. A MATLAB implementation has an advantage over a FORTRAN implementation that its environment allows for an easy *what-if* analysis. An evaluation by this author is that since, in the basic algorithm, most of these steps occur sequentially, an implementation in *conventional programming* language can save valuable time. Further, looking into the algorithm, a number of book keeping activities have been noted which are best carried out by using a conventional programming language. The search process for the best controller is the only part which merits an AI type language. Here using heuristics and simple evaluation functions, the search was pruned and guided.

In addition to the above improvements, in this thesis, the developed system (the VVCM) has the following enhancements:

- The expert system takes into account weather and seasonal information which allows computation of the limits of the system in a dynamic fashion.
- A neural network based system predicts the new power flows over the transmission lines and transformers that emerge based on the new settings of the controllers.

2.4 Linear Programming for Voltage/VAr Violation Problem

This section documents earlier attempts in the literature, to manage voltage and reactive power using purely analytical techniques.

[Mamandur81] presents a mathematical formulation of the optimal reactive power control. The method minimizes the real power loss in the system. The method employs linearized sensitivity of power system variables to establish both the objective function for minimizing the system losses and system performance sensitivities relating dependent, and control variables. Then it uses a dual linear programming technique to determine the optimal adjustments to the control variables, simultaneously satisfying the constraints mentioned above. Results have been presented for a six bus system.

---

[2] an expert system shell

[3] "Artificial Intelligence Aided Voltage Stability Enhancement," *Canadian Conference on Electrical and Computer Engineering*, May 26th - May 29th, 1996, Calgary, Alberta, pp. 392-395.

[Mamandur82] presents a method to determine the minimal adjustments to the reactive power (VAr) control variables required to alleviate over-voltages, under-voltages, and generator VAr limit violations. As the method minimizes the control adjustments to the control variables, it is particularly suitable for emergency situations. Results are presented for a 6-bus system and the modified IEEE 30-bus system for different loading conditions.

[Matsuda88] presents a case study for applying a distributed control scheme to power system control. Combined injection of VAr compensating devices controlled by a distributed expert system has been proposed as a measure to maintain voltage stability in heavy loading conditions. A simulation study has been carried out using five workstations which represent a power system and four VAr compensating devices. The results demonstrate the effectiveness of the proposed system for voltage recovery. [Matsuda88]. According to the authors, the utilization of experiences and heuristics by rules are not included in this application because the control rules for VAr injection and communication between the units are very simple for the particular system considered. Thus in this sense, the system might be called a rule-based system rather than an expert system.

[Mota-Palomino86] presents a linear programming based algorithm to solve reactive power dispatch problems. The penalty function-linear programming technique is used to obtain a complete reactive power dispatch solution algorithm. A systematic approach is proposed for the formation of sparse reactive power sensitivity matrices. Results are presented for a 256 bus system and 58 control variables.

[Sun85] presents a methodology for voltage and reactive power dispatch that has been developed to enhance a system's capability to cope with outages. In this method, not only the base case but also associated contingency cases have been considered. Therefore, it is referred to as the Global VAr Dispatch method. This method is developed by extending the reactive power part of the decoupled optimal load flow program and by incorporating outage-contingency constraints into the scheme to give an optimal steady-state secure operating point. The reactive power control variables are optimized according to the defined objective, subject to constraints for both the base case and under selected contingency system operating conditions. The global VAr dispatch method is formulated using the decoupled approach, with the reduced gradient method. The reciprocity theorem and linearized network models are used to achieve an efficient computational feature.

2.4.1 Comments

Based upon review of linear programming techniques to voltage/VAr and voltage stability problems, in the literature, the following improvements are being suggested, in this thesis:

- The linear programming formulated in this research allows the controllers to be moved in both directions.
- In the literature, only [Hsu92] applies ILP to the voltage/VAr violation problem. However, the authors have ignored the action of a generator as a controller. Generally, to solve an ILP/MILP problem, the *branch and bound* method is applied [Rekliatis83]. This method is very computationally intensive. Very often for problems with a large number of variables this method fails. In this work, a method called *discretize and eliminate* [Grace93] has been tried.

2.5 Voltage Stability

2.5.1 Background

Electric utilities have been forced in recent years to squeeze maximum possible power through existing networks due to various limitations in the construction of generation and transmission facilities. Voltage stability is concerned with the ability of a power system to maintain acceptable voltages at all nodes in the system under normal and contingent conditions. A power system is said to have entered a state of voltage instability when a disturbance causes a progressive and uncontrollable decline in voltage.

The problem of voltage instability (or voltage collapse) can be traced due to the inability of the power system to supply the reactive power (to the system) or by an excessive absorption of reactive power by the system itself.

The phenomenon of voltage collapse is characterized by a progressive decline of voltage in one or more consumer regions, a fall which accelerates after a few minutes and is typically accentuated by the action of on-load tap changers. This often happens due to the operation of the power system close to the maximum power transfer capability (stability limit). This problem usually occurs due to disturbances or significant difference between load forecasts and actual loads. In order to preserve the security of the system, during an unusually severe situation, meaningful precautionary measures are taken. These include the locking of transformer on-load tap changer mechanisms.

The IEEE working group [IEEE93] on voltage stability has given the following definitions:

- **Voltage stability** is the ability of the system to maintain voltage so that when load admittance is increased load power will increase, and that both power and voltage are controllable.

- **Voltage collapse** is the process by which voltage instability leads to loss of voltage in a significant part of the system.

- **Voltage security** is the ability of a system, not only to operate stably, but also to remain stable following any reasonably credible contingency or any adverse system change.

The above IEEE group has identified the following factors leading to voltage instability:

- Load increase (real or reactive)
- Loss of a generator (not necessarily loss of real power)
- Loss of shunt compensation
- Drop in sending end voltage
- Loss of transmission lines.

In the case of a *voltage stability* problem, one may have to resort to load-shedding.

2.5.2 Load-Shedding

One argument for under-voltage load shedding is that load will be lost anyway during abnormal voltage - because of contactor dropout, or electronic power supply shutdown. It is better to keep the load shedding under utility control, with known trip settings and time delays.

Voltage stability problems will probably be with us for the foreseeable future. Under-voltage load-shedding is one of the lowest-cost solutions to voltage stability problems. While this method provides low cost protection against voltage collapse, hopefully this is not the ultimate answer [Taylor92]. There is natural reluctance to expose customers to unnecessary interruptions. Other measures being evaluated (which take longer to implement) include more shunt and series capacitors, static VAr compensation, and new transmission lines. The role of under-voltage load-shedding may be limited to protection against disturbances outside deterministic planning criteria.

Advances in technology in predicting impending voltage collapse, in distribution automation, and in computers and communications, may provide future alternatives.

2.6 Expert Systems for Voltage Stability

As compared to the voltage/VAr violation problem, the area of voltage stability has received less attention from the community of researchers applying artificial intelligence to power systems.

2.6.1 An Intelligent Load Shedding Approach to Enhance The Voltage Stability of a Power System

[Vadari91] describes an expert system for *load-shedding* to maintain *voltage stability* called the ELS (Expert Load Shedding). The steps in the algorithm are:

1. **Preliminary Steps:** Divide the load at each bus into smaller loads and assign a priority number for each load.
2. **Area Identification:** Low voltage areas are identified. A low voltage area comprises a set of connected buses (within the same electrical island) all of which have voltages lower than a pre-defined threshold voltage. Also, this low voltage area is completely surrounded by buses whose voltages are higher than the threshold. This program module is written in the *C* language.
3. **Reactive Power Mismatch Calculation:** Estimate the reactive power mismatch for the buses with low voltage problems. It is assumed that if this amount of reactive power is shed at the bus with the lowest voltage, the voltage at that bus will rise above the threshold. This is done using the following algorithmic procedure:

Reactive power mismatch at bus x =
(Voltage at bus x - Threshold voltage)/ (Sensitivity Factor (γ))

where the sensitivity factor γ is calculated as follows:

Powerflow equation:

$$[J]\begin{bmatrix}\Delta\theta \\ \Delta V\end{bmatrix} = \begin{bmatrix}\Delta P \\ \Delta Q\end{bmatrix} \tag{2.2}$$

where

J = Jacobian matrix
θ = Bus voltage angle
V = Bus voltage magnitude
P = Bus real power mismatch
Q = Bus reactive power mismatch.

Rearranging equation 2.2,

$$\begin{bmatrix}\Delta\theta \\ \Delta V\end{bmatrix} = [J^{-1}]\begin{bmatrix}\Delta P \\ \Delta Q\end{bmatrix} \tag{2.3}$$

Therefore, if bus x is perturbed by 1.0 MVAr, and
$\Delta P_i = 0.0, \forall$ i = 1 to n
$\Delta Q_i = 0.0, \forall$ i = 1 to n, $i \neq x$, $\Delta Q_x = 1.0$.

Therefore, the calculated value of change in voltage (ΔV_x) at bus x due to this 1.0 MVAr perturbation at the same bus is determined to be γ.

4. **Load Shedding:** In this mode, load shedding takes place. In the ***automatic load shed mode*** enough load is automatically shed to satisfy the reactive power mismatch of the low voltage area as calculated by the reactive power mismatch module. In the ***operator assisted load-shed mode*** the program presents the information gathered by the above two modules and the operator decides which loads should be shed. If desired, this module can identify the load to be shed for the operator.

2.6.2 Rotor Heating Criteria

Voltage stability hinges on the continuing control of the total system's supply and demand of reactive power within the overall transmission, sub-transmission and distribution networks which connect the power stations to the consumers. This is complicated by the elements interspersed within this extensive network that both supply and consume reactive power. Extensive collapses have followed disturbances more severe than those considered at the planning stages, but there is a puzzling delay of minutes between the disturbance and the collapse. By simulating the physical process, the critical factor has been found to be the action of rotor over-current protection in curtailing the generator reactive power output. The puzzling delay has been solved by the mathematical modeling of the generator rotor's thermal response which shows that the above rated rotor current would take minutes to reach the rotor's temperature limit.

[Lachs95] describes heat run tests for rotors to obtain parameters for mathematical rotor thermal models for generators of sizes 500 MW and 600 MW. These tests have been used to devise valuable tools to counter voltage instability.

2.6.3 Derivation of Rules from Rotor Heating Criteria

The above study has led to the formulation of rules for an expert system [Lachs90] to deal with voltage instability. The algorithm that has been formulated is both simple and reliable and allows computer evaluation, in a fraction of second, of the time for the rotor temperature to be reached. When integrated into SCADA, this information can give an early warning and so provide a valuable tool to control system voltage stability. At present there are 13 rules in this knowledge base module. Six of them are directly derived from the emergency control strategy, based on rotor heating criteria. It may be mentioned that these rules are for a specific power system.

1. Rule 1: The reactive power on any hydro or synchronous condenser unit should not increase by more than 30 MVAr, otherwise its terminal voltage will be raised to 106%;

2. Rule 2: No load bus should experience a voltage reduction greater than 1.5% from normal, otherwise load transformer tap ratio will be adjusted by 1%.

3. Rule 3: The reactive power on any thermal generator should not increase by more than 30 MVAr, otherwise the thermal generator transformer tap ratio will be adjusted by three 1% steps.

4. Rule 4: The voltage reduction on any load bus should not be less than 1.5% following action due to rule 2 being violated, otherwise the load transformer tap ratio will be adjusted for the second time by another 1%.

5. Rule 5: The voltage reduction on any load bus should be less than 1.5% following action due to rule 4 being violated, otherwise the load transformer tap ratio will be adjusted for the third time by another 1%.

6. Rule 6: The voltage reduction on the load bus should now be less than 3% after the above steps, otherwise load will be shed.

2.6.4 Comments

[Vadari91] in his expert load shedding (ELS) program bases his calculations on an algorithmic method and uses production rules for deciding the priority of the loads. It does not take into consideration what happens to the rest of the system. [Lachs90] utilizes the capabilities of the generator to prevent voltage instability but does not take into account a complete view of the voltage stability problem. *To counter this problem, in this thesis, three load-shedding modules have been formulated which also predict what happens to the rest of the system, for contemplated load shedding.*

2.7 Miscellaneous

2.7.1 Line Flows

[Yang94] presents a machine learning method to estimate the line flows and bus voltages following an outage event. A decision tree which is capable of generating the desired line flows and bus voltages is created using the training patterns which are compiled from historical operating records of the Taiwan power system. The established decision tree contains all the essential knowledge for line flow and bus voltage predictions. Some of the limitations of this technique are:

- A tree is needed for every load bus.
- The size of the decision tree is directly related to the discriminating power of the attributes. The more powerful is the classifying ability of the attributes, the smaller the decision tree will be. This demands much experimentation.
- The results are sensitive to noise or bad data in the various training patterns.

2.7.2 Voltage Instability Prediction

Although voltage stability is a dynamic phenomenon, which may involve slow varying controllers, the vast majority of the reported research activities deal with the static aspects of the problem only. [IEEE93] reports about the use of four indices which can indicate the proximity or nearness to singularity of the static Jacobian matrix. The latter will singular when Glavitsch's [Kessel86] index $L=1$, or Tamura's [Tamura83] $VIPI = 0$, or Schlueter's [Schlueter87] $S = 0$, or Thomas's [Thomas86] $\sigma_{min}(J) = 0$. The determinate of the Jacobian matrix is zero for all these cases and the system is at a bifurcation point.

According to [IEEE93] the index L is based on single node calculations and is therefore easy to calculate. It does however require substantial engineering judgment in a large system. The multiple solution uses a sequence of calculations, including the measure $VIPI$ to track low voltage equilibria, one stable and the other unstable. The method is physically based, easy to calculate and provides local information on voltage sensitivities as well as system wide measures. Sensitivity based indices, however are extremely nonlinear which makes it difficult to asses the proximity to voltage instability. The efficiency of this method in large systems has not yet been proven. Thomas's index views the entire Jacobian matrix and uses a computationally sound norm-based index to indicate proximity to singularity. While computationally sound (meaning it does not lead to computational or algorithmic difficulties in a rank deficient matrix problem as is the case with eigen value computations), it has been noted that the Jacobian matrix may become singular for reasons other than voltage problems. Therefore, the singular value may be a more generic index, but this index is harder and more time consuming than the previous two techniques. Schlueter determines voltage control areas and evaluates an area version of the minimum eigen value index. It is computationally more efficient. However, the approximations make it difficult to rationalize the techniques without further extensive testing.

2.8 Conclusion to Chapter 2

Based on the above literature review and the already identified research objectives, the work in this thesis will focus on development of the following modules:

1. An **E**xpert **S**ystem **M**onitor (ESM) program to find voltage/VAr violations and their severity.

2. A module called the **V**oltage **Q** **S**ensitivity **M**ethod (VQSM) that decides whether the system is voltage stable or not (based on the reduced Jacobian matrix).

3. A program that is able to classify the current power system operating mode into ***stable***, ***intermediate stable*** or ***unstable mode***, based on the power system data. This

classification system will be built using an induction decision tree algorithm (a data mining methodology.)

4. A module called the **V**oltage **VA**r **C**ontroller **M**ethod (VVCM) that makes decisions to remove voltage/VAr violations in such a way that the number of changes in the settings of the controllers are minimized.
5. A supplemental tool to the VVCM, based on linear programming and mixed integer linear programming concepts, for severe violations where the VVCM may fail or be inefficient (may fall into infinite loop).
6. A module called the **L**oad **S**hedding **M**ethod (LSM) that will make decisions on where and how much load is to be shed.
7. A neural network based program which will predict the new power flows over the transmission lines and transformers, once the settings of the controllers have been changed.

Chapter 3

ENABLING TECHNOLOGIES

One of the objectives of this thesis is to investigate the applicability of artificial intelligence and its sub-fields (expert systems, data mining and artificial neural networks) for providing assistance to power system operators for making decisions. This chapter gives the pertinent background to the enabling technologies that have been used to achieve the objectives of this thesis.

3.1 Artificial Intelligence

The study of artificial intelligence as a branch of computer science started four decades ago. A team led by John McCarthy of M.I.T. mooted the idea of thinking machines. Although there is no universally accepted definition of artificial intelligence, the following definition purports to draw an approximate boundary around the concepts of artificial intelligence: *"Artificial intelligence is the study of how to make computer do things which at the moment, people do better"* [Rich91]. Reference [Nilson90] explains this as follows: Over the past few decades several computer systems have been built that can perform tasks such as writing computer programs, doing mathematics, engaging in common sense reasoning, understanding language, and even driving an automobile. These activities are examples of human mental activities. Additionally, there are computer systems that can diagnose diseases, plan the synthesis of complex organic chemical compounds, solve differential equations in symbolic form, analyze electronic circuits, understand limited amounts of human speech and natural language text, or write small computer programs to meet formal specifications. These systems are said to possess some degree of artificial intelligence.

3.2 Expert Systems

An expert system is a computer program that represents and reasons with knowledge of some specialist subject, with a view to solving problems or giving advice [Jackson90]. Thus expert systems are software packages which substitute human expertise by computer programs. These computer programs contain within them the knowledge of the expert(s) in a particular field in a codified form. Portability of software makes the use of expert systems very attractive where human expertise is scarce or costly. Expert systems are characterized by storage of huge quantities of knowledge bases.

An expert system manipulates vast amounts of information with the intention of solving a particular application problem. What makes expert systems unique is the way they approach and solve problems (combining knowledge representation and problem solving algorithms) and the different types of problems they can solve (heuristic versus conventional problems). Expert systems attempt not only to apply conventional mathematical and Boolean operators, but also incorporate typical human reasoning processes such as *rules of thumb* and *shortcuts* to solving problems. The goal of an expert system is to mimic an expert's thought process in solving a problem.

Expert systems have the capacity to manipulate problem statements and integrate relevant pieces of knowledge from a knowledge base (a collection of information) using reasoning techniques commonly known as heuristics, to emulate the expert. By their very nature, expert systems address and solve knowledge-intensive problems that can have many correct or acceptable answers.

The following information on expert systems has been derived from [Jackson90, Nilson90, Rao91, Rao93, Rich91].

3.2.1 Characteristics of Expert Systems

The typical characteristics of the problems which are tackled using expert systems are as follows:

1. Problems having a large solution space with no particular correct answer but a number of possible answers each having its plus and minus points.
2. Problems that involve uncertain and/or incomplete data.

3. Solution procedures which cannot be laid into straightforward sequential algorithms and hence demand educated guesses (heuristics) based on past experience.
4. Problems whose solutions demand a large amount of knowledge.

3.2.2 Components of Expert Systems

The main components of an expert system are:

1. **The knowledge base** contains expert knowledge in codified form. The knowledge base of an expert system houses the information used by the expert system in pursuit of a solution to a problem. A knowledge base contains static data, as in a conventional database, and relational information. This includes the related and unrelated relationships of the data of the problem and its domain. Production rules, the basis of most expert systems are located here. The knowledge base has two parts:
   (a) **The database** of the knowledge base includes the facts of the problem, both related and unrelated. This is a passive area of the expert system - simply storage device for data and formulae. The information includes the given and unchanging knowledge about the problem and its domain.
   (b) **The working memory** of the knowledge base is modified by the inference engine as situations and data change. Working memory is used only during processing and is the resident space for information processing. Working memory takes data from the data space, knowledge from the knowledge base, and combines them with the information supplied from the user to be processed by the inference engine in pursuit of a solution.
2. **The inference engine** is the workhorse of the expert system. It decides (based on input data or query) as to which information from the knowledge base is to be used for solving the problem on hand and in which order. It consists of the processes that work the knowledge base, do analysis, form hypotheses, and audit the processes according to some strategy that emulates the expert's reasoning. The inference engine processes new information, combines it with the knowledge base, considers the relationships in the knowledge base, and proceeds to solve the problem in the working memory using its established reasoning and search strategies. Hence, one may say that the inference engine is the *thinker* of the problem solving system and provides the overall control.

3. **Interfaces** include terminals, graphical representations, multiple graphic windows, etc.. These are specifically useful for user interaction with the expert system so that one can enhance the knowledge base or get answer to problems.

The strength and power of an expert system lies in its knowledge base. Expert systems are usually successful if they are rich in knowledge, even if they have poor problem solving capabilities. This knowledge is largely heuristic - judgmental, experimental and uncertain. In summary, expert systems are complex programs to construct and maintain, but in return offer tremendous problem-solving power and efficiency.

3.2.3 Heuristics

Heuristics are rules of thumb to improve the efficiency of search programs by making short cuts. They are based upon the expert's knowledge, experience and intuitive feeling. An application of heuristics ensures that a good workable solution is obtained in a reasonable amount of time. This may not lead to an optimal solution, but if a systematic and a sequential search for the best (optimal) solution is made, it can be very time consuming and in many cases impractical.

3.2.4 Knowledge Engineer

A knowledge engineer is a person who performs the following functions:

1. Defines the expert system domain.
2. Elicits desired knowledge from human experts.
3. Codifies the knowledge into the knowledge base.
4. Designs the inference engine, if the tool used for building the expert system does not have one.
5. Tests the system to evaluate its robustness and accuracy.

3.2.5 Knowledge Representation Models

Knowledge representation models describe the various architectures used to represent the expert's knowledge in an organized and consistent manner in the knowledge base. Knowledge is represented commonly in expert systems by one of the following methods:

1. **Semantic networks** are collections of objects (commonly referred to as nodes) that are linked together by a relationship. Nodes represent objects, concepts, and situations and they are shown graphically by dots, circles or boxes. Relationship between objects is represented graphically by an arrow between the two nodes for which the relationship is intended to be expressed. The basic construct of semantic networks is easy to understand. However, they can rapidly become very large and complex.

2. **Frames** consist of a collection of slots that contain an attribute to describe an object, class of objects, a situation, an action, or an event. Frames provide a concise, structural representation of useful relations that capture the way an expert typically thinks about the data in the knowledge base. The primary advantage of frame representation is that it is more concise and compact. Thus less time is required in searching for specific information. For example, consider the objects of a substation such as breakers, switches, buses, transformers, and transmission lines. The frame representation is given below:

```
SUBSTATION FRAME:
Name:  character string
Breakers: (pointer to breaker frame)
Bus sections: (pointer to bus section frame)
Lines: (pointer to line frame)
Transformers: (pointer to line frame)
```

   The frame structure allows an expert system to facilitate data management both at run time and during maintenance intervals.

3. **Production rules** are conditional *If Then* or *If Then Else* description of a given situation or context of a problem. Production rules are of the format:

   IF (premise(s)) THEN (action_1) ELSE (action_2)

A rule is said to be activated when all of the conditions in the premise of a rule are satisfied by the current situation. The production rule is said to be fired (or executed) when the action is performed. The following rule is an example:

If the voltage at a bus is below 0.95 p.u. inform the operator that the power system is operating in an alert state.

Rule based systems have the advantage of being modular, uniform and natural. However this uniformity may introduce a rigid structure. Since every action must go through the match and action cycle, it may be inefficient in some cases.

4. **Predicate Calculus** or logic programming, relies on the truth and rules of inference to represent symbols and their relationships to each other. For example consider the rule *If A, Then B*. Following this rule if *A exists*, then *B* also exists, can be inferred. This provides a simple way of determining the truth or falsity of a statement. Predicate calculus can also be used to represent statements about specific objects or individuals. The form of logic most often used is the first order predicate logic (FOCL), which is an extension of predicate calculus. First order predicate logic works with variables, predicates, sentences, connectives, qualifiers, and questions. Predicates describe a relationship or make statements about the variable under consideration. For example, the following PROLOG statements might be used in an expert system to guide power system operators in system restoration:

```
restoration_requires(X, yes)
 Component(X),
 charge_state(X,Y), = / = (Y, is_charged),
 fault_state(X,Z), = / =(Z, is_faulted).
```

This example states that in order for object X to have a restoration requirement of *yes*, it must be a component which is not charged and not faulted. Predicate calculus has two major advantages:

   (a) **Preciseness:** Logic is a precise, standard method of determining the meaning of an expression.

   (b) **Modularity:** Statements can be added, deleted, or modified without having to consider the impact on other statements.

Each knowledge representation model has its advantages and disadvantages. The objective is to select a knowledge representation model that closely mimics the real world application. For each particular problem, the expert already has a predefined structure or conceptual level of the domain, even though it may not be explicit. The selection of a knowledge representation model closer to the expert's would lend itself to easier coding of the knowledge.

3.2.6 Problem Solving Strategies

The inference engine of an expert system must have some way of traversing, searching, or reasoning through the knowledge base to logically deduce, infer or prove why a problem occurred. It must know where to start solving a problem and be able to resolve any conflicts that might arise during the problem-solving process. The inference engine attempts to emulate the way human experts would approach the same class of problems. Searching is the key to the problem solving process. If the path is unsuccessful, alternative paths are tried in search of a solution. The method of solving a problem is similar to that of solving a maze: an expert starts with the given solution and tests alternatives until a solution to the problem is found. Alternatives are defined as paths or steps/ actions taken to traverse through the domain (that is the knowledge base). The goal of problem solving is to find the best search techniques, namely those which will locate a solution path efficiently and effectively by limiting the number of data items (paths) examined. Problem solving can be divided into three steps:

1. **General Search Techniques:** These are the foundations of many problem solving techniques. Searches can be divided into two categories: blind searches and heuristic searches. Blind searches are referred to as *weak* searches, weak in the sense that they employ no intelligent decision making in the search. Blind search can be very costly if a large knowledge base has to be explored. Heuristic searches are referred to as *strong* searches. A heuristic search involves the analysis of domain specific information, thereby reducing the search space.

2. **Control Strategies:** These guide the direction and execution of the search. Different situations require different reasoning techniques for deciding what to do next: that is, deciding which operation to apply and where to apply it. Forward chaining, backward chaining, and bi-directional chaining are some of the control strategies used in expert systems.

3. **Additional Reasoning Techniques:** Problems in the real world are not necessarily solved using general search methods. Various techniques such as disjunctive facts, certainty factors, constraints, abstractions, fuzzy logic, black boarding, and guessing, assist in the search of a quick and efficient solution path.

3.2.7 General Purpose AI Languages

Every expert system is developed using some type of programming language, although most employ a general purpose AI language. The most common AI languages for expert systems are LISP and PROLOG. Several expert systems have been developed in non-AI languages (PASCAL, FORTRAN and C), primarily for ease of maintenance. The advantage of developing a system using an AI language is that it provides for symbolic processing. The disadvantage is that it requires another level of expertise; that is, an individual who is proficient in an AI language programming.

3.2.8 Expert System Development Tool (Shell)

Expert system development tools are *off the shelf* skeleton expert systems that have been explicitly designed to assist in building expert systems. Also known as *shells*, they comprise essentially a predefined inference engine that knows how to use the knowledge base to reach conclusions. An advantage of starting with the development tools is that knowledge acquisition tools and utilities have already been built. One can use the shell to build an expert system, without having to create the reasoning and data structures components from scratch. The primary disadvantage of a development tool is that it will generally employ only one reasoning methodology and knowledge representation technique, while many applications often require a combination of techniques. Some of the well known shells available are KEE, RULEMASTER, EXSYS, ADVISOR, ART, CLIPS.

3.2.9 Applications of Expert Systems

Some well known expert systems are:

1. **MYCIN** is developed to diagnose and treat infectious blood diseases.
2. **R1** is designed to configure orders for Digital Equipment VAX computers.
3. **PERSONAL CONSULTANT** is designed by Westinghouse to assist it in its research on metal corrosion.
4. **Hearsay - II** employs multiple expert systems to understand connected speech in a thousand - word vocabulary.
5. **Interist** is an expert system to make diagnosis in internal medicine.

3.2.10 Expert System Implementations in EMS

There are three potential approaches for integrating expert system software into the EMS computer [Sagakuchi88].

1. **Expert System in Numeric Programming Languages:** The expert system has to be implemented in a numerical programming language such as FORTRAN, PASCAL or C unless the knowledge programming environment is supported. This method will work well when the problem is small, but will fail for a large scale problem typical of power system operation.
2. **Expert System in AI Programming Languages:** If the real time operating system supports an AI language, the expert system is implemented in the same way as a conventional program. This method may result in slow performance. One way to avoid this is to implement the inference engine in another language, for example the C language.
3. **Expert System in Attached Computer:** In this alternative, a dedicated processor for the expert system is linked to the EMS computer.

3.2.11 Comments

The benefits of successful introduction of expert systems to power systems may be summarized as follows:

- A high level of expertise would be continually available, irrespective of staff changes, mobility, etc..
- Every requirement or situation would be examined with equal thoroughness and objectivity.
- It can relieve the operator of routine decision making.
- Once included in the system, new knowledge would be fully available with less detailed updating or training necessary.
- The expert system can provide intelligent tutoring.

Expert systems have been used as valuable support tools for different facets of power system operations [Madan97b][1], [Dillon90, Zhang89]. However, expert systems have some limitations. The most important concern is their reliability. Expert systems are limited by the information in their databases and by the nature of the process for putting that information in. They cannot report conclusions that are not already implicit in their databases. The trial and error process by which knowledge is elicited, programmed, and tested, is likely to produce inconsistent and incomplete databases; hence an expert system may exhibit important gaps in knowledge at unexpected times. Moreover, expert systems are unlikely to have complete, clear functional specifications, and their designers may be unable to predict reliably their behavior in situations not tested. If the databases are not structured as modules, it may be difficult to modify and maintain them.

One of the prime difficulties in developing expert systems is the problem of capturing the expertise of human experts. This problem is called the Knowledge Acquisition Barrier. Some of the reasons for this knowledge acquisition barrier are:

- Experts find it difficult to articulate their expertise.
- The inability of the knowledge engineer to ask pertinent questions from the domain engineer to extract relevant information.
- Experts are afraid of passing on their expertise because of the fear of losing their importance.

In this context, the use of machine learning and data mining algorithms, as a mitigating solution to the problem of the knowledge acquisition barrier, has become useful.

3.3 Machine Learning

Learning can be defined as *"any change in a system that allows it to perform better the second time, on repetition of the same task or another task drawn from the same population* [Simon83]. The subject of machine learning is concerned with the study and the computer modeling of the learning process. Machine learning systems can be classified along various dimensions. One scheme uses the amount of inferencing a machine learning program must do [Carbonell83]. At one end of this continuum, there are *rote learning* programs which learn by direct encoding of the knowledge. A learning system that performs work based on lookup tables is an example of a rote learning system; e.g., a database system. At the other end of this continuum are *unsupervised learning* approaches which attempt theory formation or classification schemes without feedback from the environment. This resembles the process of scientific discovery. An example would be the discovery of a new law or relationships without guidance from a teacher or textbook. In between these two extremes on the above continuum are learning classification rules from examples of different classes or categories. This is a form of *inductive learning from examples*. An example for this type of learning system is determining how to classify chemical elements in a periodic table.

---

[1] "Applications of Artificial Intelligence for Power Systems," *Electric Power Systems Research Journal*, vol. 41, no. 2, May 1997, pp. 117-131.

3.3.1 Induction Learning from Examples

Learning involves generalization from experience. The performance should improve not only from repeating the same task but also on similar tasks in the domain. It is observed that the interesting domains tend to be large, and a learner can examine only a fraction of all possible cases. From this limited experience, the learner must acquire knowledge that will allow generalization of unseen instances of the domain. Induction learning from example takes place when the input is in the form of (x_i, y_i), and the program is expected to learn a function f, such that $f(x_i) = y_i$, for all i. Further, the function f should capture general patterns in the previously unseen input x values. x_i is a composite description of some object, situation or object (for example, a faulty machine), and each y_i is a simpler description (for example, the problem of that machine). This is also called supervised learning, because the x_i values are typically provided by a supervisor or teacher. In this paradigm the learner is provided with a set of positive and negative examples.

Inductive learning can be of two types [Jackson90]:

1. **Concept learning** occurs when a learner is provided with sets of data, some of which are examples of a concept and some of which are not (the counter-examples). The learning task is to find a concept that includes all of the examples and none of the counter-examples. For example, consider the data set: {Air Canada, Canadian Airlines, Canada 3000, United Airlines, American Airlines, Delta Airlines}. If positive examples are {Air Canada, Canadian Airlines} and negative examples are {United Airlines, American Airlines} then a plausible concept that covers all the examples but none of the counter-examples would be *an airline company whose prime owners are Canadians.*

2. **Descriptive generalization** occurs when a learner is given the class of objects that are positive examples representing the concept, and the learning task is to derive a description that specifies the properties of objects belonging to that class. For example, if the given concept is *an airline company* with members {United Airlines, Canada 3000} then an appropriate descriptive generalization is: *a company whose business is transportation of people and cargo by air, and which has a fleet of aircraft.* These descriptive concepts should be provided in the form of background knowledge in the system.

In concept learning, one is given a set of positive examples and negative instances of some concept, and the task is to generate a rule that will recognize unseen instances of this concept. In descriptive generalization, one is given a set of instances and the class to which they belong. The task is to derive the most parsimonious description which may be applied to each member of the class.

3.3.2 Data Mining

Data mining[2] is defined as the nontrivial extraction of implicit, previously unknown, and potentially useful information from data [Frawley92]. This field focuses on the classification of *new*, *useful* and *interesting* relationships that may exist within the large archives of information.

The discovery of relationships within archived data is a potentially powerful application of machine learning algorithms. This sort of work has its origin in efforts to automate knowledge acquisition for expert systems - why ask an expert for decision rules if they can be induced from archived data? In fact, this approach was taken in the eighties by Michalaksi and Chilausky [Michalski81] to induce classification rules for diagnosing soybean diseases, using the archived cases. The rules induced across cases were more accurate at the classification of disease than those provided by human experts.

Clearly, the potential to identify new predictive relationships between attributes from large databases is important. For example, oil exploration companies archive complex quantitative and qualitative data for drilling sites. In the finance sector, discovery from databases may be used to make credit approval, bankruptcy prediction, stock market prediction, security fraud detection, mutual fund selection, etc.. In the medical field the potential exists to discover drug side effects, hospital containment costs and genetic sequence analysis [Frawley92]. In general, analyzing archived data for predictive or characteristic relationships may generate knowledge of considerable potential significance. Data mining is an extension of the machine learning algorithm. The relationship between machine learning and data mining is given below:

---

[2] This notion of discovery has been given various names, including knowledge extraction, database exploration, data pattern processing, data archaeology, information harvesting, software and data dredging. It is also known as *Knowledge Discovery in Databases* where the discovery algorithms focus on data stored in databases, such that they can exploit the inherent opportunities presented by organization of data in databases.

3.3.3 Data Mining and Machine Learning

Data mining and the part of machine learning dealing with induction learning from examples overlap in the algorithms used and in problems addressed. The main differences are:

- Data mining is concerned with finding *understandable* knowledge, while machine learning is concerned with improving the performance of a learning agent.
- Data Mining is concerned with very large, real-world databases, while machine learning *typically* (but not always) looks at smaller data sets. Thus, efficient learning questions are much more important for data mining.
- Machine learning is a broader field which includes not only learning from examples, but also reinforcement learning, learning with a teacher, etc..

Therefore, one can say that data mining is that part of machine learning which is concerned with finding *understandable* knowledge in large sets of real-world examples.

3.3.4 Data Mining Tools

A number of tools have been developed to carry out data mining. The URL *http://info.gte.com/* lists some of these tools. Data mining deploys a variety of algorithms. Generally, the more algorithms in use, the higher the likelihood of accurate results. The idea is that if one algorithm cannot find something, then another will. As a result, data mining vendors are fielding products that combine algorithms based on neural networks, induction, association, fuzzy logic, numerical (or statistical), and visualization. In whatever combination they are employed, the algorithms attack problems using one or more of the following approaches:

1. **Classification Approach** may take the form of a *Decision Tree Approach*, e.g. ID3, C4.5, AC2; *Neural Network Approach* e.g. MATLAB Neural Network Toolbox; *Rule*

Discovery Approach e.g. DBMiner, CN2, IDIS; *Genetic Programming Approach* e.g. OMEGA; *Nearest Neighbor Approach* e.g. PEBLS.

2. **Clustering tools** are used to segment a database into subsets or clusters, with the members of each cluster sharing a number of interesting properties. A cluster is a smaller and more manageable data set that can be presented to the supervised inductive learning program for classification. Examples of these tools are: SNOB, AUTOCLASS III.
3. **Dependency Derivation** tools are programs that assist in the construction of causal explanations for sample data and their use in prediction, e.g. TETRAD II.
4. **Deviation Detection** tools are programs that use extracted patterns to find anomalous or unusual data, e.g. EXPLORA.
5. **Summarization tools** deal with finding patterns that relate groups of instances (inter-record dependencies), e.g. EMERALD.
6. **Visualization tools** provide analysts with visual summaries of data from a database. It can also be used as a method for understanding the information extracted using other data mining methods. Features that are difficult to detect by scanning rows and columns of numbers in databases, often become obvious when viewed graphically. An example is IBM's Data Explorer.
7. **Statistics tools** use statistics to find new and interesting patterns, e.g. JMP, SAS.
8. **Multistrategy tools** are a collection of data mining techniques, e.g. MLC++.

Development of these tools has given an impetus for the application of data mining to many areas. As already stated, one way of classifying the above algorithms is the amount of inferencing a machine learning program must do [Carbonell83]. For real discovery to occur, only programs capable of making unsupervised discovery could qualify as true data mining techniques. However due to limited success with these programs, supervised discovery programs have also been referred to as data mining programs. Examples of supervised and unsupervised discovery algorithms are given in the next two sections.

3.3.5 Example of Supervised Discovery

The term ID3 algorithm [Quinlan86] which stands for *Induction of Decision Trees* is a supervised learning system that constructs decision trees from a set of examples. ID3 uses an iterative method to build up decision trees, preferring simple trees over complex ones, on the theory that simple trees are more accurate classifiers of future inputs. An induction tree is constructed as follows:

ID3 Algorithm

1. *An attribute is selected as the root of the tree. Branches are made for all different ways this attribute can have.*
2. *The tree is used to classify the training set. If all examples at a particular leaf belong to the same class, this leaf is labeled with this class. If all leaves are labeled with a class, the algorithm terminates.*
3. *The process for the root node is repeated for non root nodes. The algorithm continues with step 2.*

Induction Decision Tree Theory

ID3 induces concepts from examples. ID3 represents concepts as decision trees, a representation that allow us to determine the classification of an object by testing its values for certain properties. One may think of each property of an instance as contributing a certain amount of information to its classification.

Shannon's information theory provides a mathematical basis for measuring the information content of a message. A message is an instance in a universe of possible messages. The act of transmitting a message is the same as selecting one of the possible messages. Therefore, one can define the information content of a message as depending upon both the size of the universe and the frequency with which each possible message occurs.

Shannon formalized these intuitions by defining the amount of information in a message as a function of the probability of occurrence of each possible message. Given a universe of messages, $M = \{m_1, m_2, ... m_n\}$ and the probability $p(m_i)$, for the occurrence of each message, the information content of each message in M is given by:

$$I(M) = \sum_{i=1}^{n}(-p(m_i)log_2(p(m_i)) \qquad (3.1)$$

The information in a message is measured in bits. For example, the information content of a message telling the outcome of the flip of a honest coin is:
I(Coin toss) = -p(head) log_2 (p(heads)) - p(tail) log_2 (p(tails))
= -0.5 log_2 (0.5) - 0.5 log_2(0.5)
= 1 bit

However, if the coin has been rigged to come up tails 75 % of the time, then the information content of a message is:
I(Coin toss) = -p(head) log_2 (p(heads)) - p(tail) log_2 (p(tails))
= -0.25 log_2 (0.25) - 0.75 log_2 (0.75)
= -0.25*(-0.2) - 0.75*(-0.415)
= 0.811 bit.

Assuming a set of training instances, C, a property P with n values as the root of the current tree, this will partition C into subsets, $\{C_1, C_2, ..., C_n\}$. The *expected information* needed to complete the tree after making P the root is:

$$E(P) = \sum_{i=1}^{n} \frac{|C_i|}{|C|} * I(C_i) \tag{3.2}$$

The *information gain* from property P is computed by subtracting the expected information to complete the tree from the total information content of the tree:

$$gain(P) = I(C) - E(P) \tag{3.3}$$

For the example in Table 3.1 if one makes the *outlook* the property tested as the root of the tree, this partitions the table of examples into the partitions $C_1 = \{1,2,8,9,11\}$, $C_2 = \{4,7,12,13\}$, and $C_3 = \{4,5,6,10,14\}$ where the sets C_1, C_2, C_3 stand for {sunny}, {overcast} and {rain} respectively.

In table 3.1 there are two classes, nine of which belong to the *Play* class and five to the *Don't Play* class.
Therefore:
I(Table) = -9/14 * log_2(9/14) - 5/14 * log_2(5/14)
= 0.940 bit

The expected information needed to complete the tree is:

$$E(outlook) = 5/14 * I(C_1) + 4/14 * I(C_2) + 5/14 * I(C_3)$$

$= 5/14*(-2/5*log_2(2/5) - 3/5*log_2(3/5))$

$+ 4/14*(-4/4*log_2(4/4) - 0/4*log_2(0/5))$

$+ 5/14*(-3/5*log_2(3/5) - 2/5*log_2(2/5))$

= 0.694 bits

S. No.	outlook	temp	humidity	windy	decision
1	sunny	85	85	false	Don't Play
2	sunny	80	90	true	Don't Play
3	overcast	83	78	false	Play
4	rain	70	96	false	Play
5	rain	68	80	false	Play
6	rain	65	70	true	Don't Play
7	overcast	64	65	true	Play
8	sunny	72	95	false	Don't Play
9	sunny	69	70	false	Play
10	rain	75	80	false	Play
11	sunny	75	70	true	Play
12	overcast	72	90	true	Play
13	overcast	81	75	false	Play
14	rain	71	80	true	Don't Play

Table 3.1: Data for Golf Playing

The information gain from this test (making outlook as the root of the tree) is:

gain(outlook) = I(from table) - E(outlook)

= 0.940 - 0.694

= 0.246 bits

Similarly, one may show that:

gain(temp) = 0.045

gain(humidity) = 0.102

gain(windy) = 0.048

Since *outlook* provides the greatest information gain, the ID3 algorithm selects it as the root of the tree. The algorithm continues to apply this analysis recursively to each subtree until it has completed the tree.

The selection of an attribute for a node is based on the *information gain* that the attribute provides. C4.5 [Quinlan93] which is an extension of ID3 has heuristics built-in to deal with large and noisy data sets. C4.5 allows nodes to be selected using the *information gain* or *gain ratio* criterion. The *information gain* criterion has a deficiency, which can be serious for a few cases. This criterion has a strong favor of tests with many outcomes. The bias inherent in the gain criterion can be rectified by a kind of normalization in which the apparent gain attributable to tests with many outcomes is adjusted. Consider the information content of a message pertaining to a case that indicates not the class to which the case belongs, but the outcome of the test. Therefore, by analogy with the definition of *info(P)* one gets:

$$split\ \ info(P) = -\sum_{i=1}^{n} \frac{|C_i|}{|C|} * log_2 \frac{(|C_i|}{|C|)} \tag{3.4}$$

This represents the potential information generated by dividing C into n subsets, whereas the *information gain* measures the information relevant to classification that arises from the same division. The concept of *gain ratio* may be defined as follows:

$$gain\ \ ratio(P) = gain(P)/split\ \ info(P) \tag{3.5}$$

C4.5 allows rules to be induced from the decision trees. C4.5 also provides for a consultation session once the decision tree has been built to classify unseen instances. When the data given in Table 3.1 *(from a machine learning database repository maintained at the University of California, Irvine)* is subjected to the C4.5 algorithm, then the results obtained are as shown in Figure 3.1.

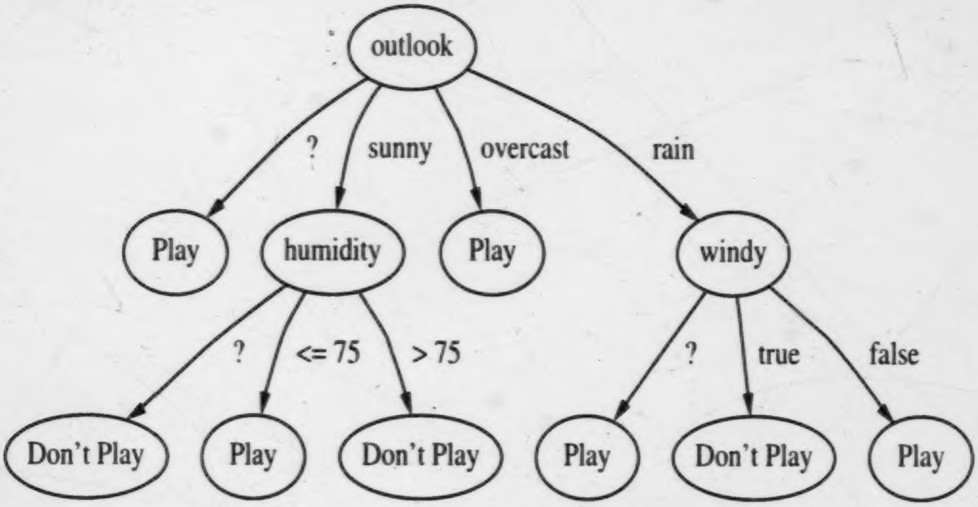

outlook
?
sunny
overcast
rain
Play
humidity
Play
windy
?
<= 75
> 75
?
true
false
Don't Play
Play
Don't Play
Play
Don't Play
Play

Figure 3.1: Induction Decision Tree for Golf Playing

It may be noted that the induction decision tree has a concept of default class shown by ? in Figure 3.1 which is useful when some data elements are missing. The following rules may be induced from the decision tree. The figure in the brackets indicates the confidence factor associated with the rules.

IF outlook = overcast
THEN class Play {70.7%}

IF outlook = rain
AND windy = false
THEN class Play {63.0%}

IF outlook = sunny
AND humidity > 75
THEN class Don't Play {63.0%}

IF outlook = rain
AND windy = true
THEN class Don't Play {50.0%}

3.3.6 Example of Unsupervised Discovery

The BACON [Langley93] system uses a data analysis algorithm to discover mathematical relationships in numerical data. It has been used to rediscover relationships such as Ohm's law for electric circuits. The training set is made up of numerical data, as generated in experiments. An example is a tuple consisting of values that are measured variables for an experiment. BACON attempts to find a relationship between the variables that hold for all examples. A relationship is a numerical expression over these variables - a term. If this term has a constant value, then it describes a numerical relationship. The search space consists of all the terms that can be constructed from the variables, using numerical operations such as multiplication, addition and division. The performance criterion is: a term is correct if it has a correct value for all the examples.

The BACON system uses a top down search: it generates a term and then checks the correctness of the term for the set of examples. If the performance criterion is not met, the term is modified. The choice of operations is guided by heuristics. An example of a heuristic is: If the value of the term and the previously constructed term increases monotonically, the ratio of the term and the variable is taken, or if the two terms vary inversely, the product is taken.

The example given below demonstrates the use of this algorithm to rediscover Kepler's third law, namely that the ratio of planet's distance to the sun d, and the period of its revolution p are related as $d^3/p^2 = k$, where k is a constant. In the table given below, p and d values are observed. Since d and p increase monotonically, the term d/p is constructed. This term is not constant and needs to be modified. Because d and d/p vary inversely, so the term d/p is modified to d^2/p. This term varies inversely with d/p, hence their product is constructed. Calculation of this term, d^3/p^2 results in a constant (with tolerance) for all examples and the algorithm stops.

Planet	p *x	d	d/p	d^2/p	d^3/p^2 *y
Mercury	0.389	87.77	0.0044	0.0017	7.64
Venus	0.724	224.70	0.0032	0.0023	7.52
Earth	1.0	365.25	0.0027	0.0027	7.50
Mars	1.524	689.98	0.0022	0.0034	7.50
Jupiter	5.200	4332.62	0.0012	0.0062	7.49
Saturn	9.510	10759.20	0.0009	0.0084	7.43

Table 3.2: Discovery of Kepler's Law

where x*= 149597870 and* $y = (1/1000000)$.

KEPLER [Wu91] is an improvement on BACON and uses a powerful algorithm called *REDUCTION*. REDUCTION enables the discovery of a complex formula with multiple variables involved by decomposing it into binary formulas with two variables at a time. The algorithm uses a set of prototype functions to facilitate discovery. An example of a law discovered from experimental data is:

$$(2 * ln(x_1) + 30.545)^3/(x_2^{1/3} + 92.248 * x_3^{1/4}) = 7.6^3 \tag{3.6}$$

The above law is derived from a set of data in well-logging, which is commonly used in oil exploration.

3.3.7 Comments

A changing business environment, moves towards a global economy, and evolving information technology are fueling the need for more capable data analysis tools. Data mining is a relatively new approach in this domain that promises increased understanding of large data repositories through the use of automated, computer-based discovery techniques. At this time, a lot of data mining work is being done using statistics. However, it is expected that in due course algorithms will be developed which will use more domain knowledge (in the form of prior or background knowledge) and less statistics. Statistical measures tend to do well at detecting differences and trends, while data mining does well at identifying relationships. In an algorithmic sense, these techniques are well understood, having had their origins in machine learning research over the last two decades. However, effective implementation and use of data mining tools still requires significant expertise in extracting, manipulating, and analyzing data from large data warehouses. Nevertheless, these tools are currently producing results that provide significant strategic and tactical advantages to businesses in highly competitive environments [Fayyad96].

As these tools continue to mature, and user interfaces improve, a larger population of analysts familiar with these tools will bring data mining to the mainstream of information technology and engineering applications, including power systems [Madan97a][3], [Madan97c][4].

3.4 Artificial Neural Networks

Artificial Neural Network (ANN) models, or simply Neural Nets, go by many names such as connectionist models and neuromorphic systems. In this respect, the ANN structure is based upon the present understanding of biological nervous systems. Instead of performing a program of instructions sequentially as in a Von Neumann computer, neural net models explore many computing hypotheses simultaneously using massively parallel nets composed of many computational elements connected by links with variable weights.

Neural network models are specified by their network topology, node characteristics, and training and learning rules. These rules specify an initial set of weights and indicate how they should be adapted during use to improve the performance of the neural network. Adaption or learning is one of the major features of artificial neural networks.

3.4.1 Artificial Neural Network

A neuron is an information processing unit of the neural network. Figure 3.2 shows the model of a neuron [Haykin94].

---

[3] "Data Mining Applications for Power Systems," *VII th International Conference on Electronics, Communications and Computers, (CONIELECOMP'97)*, Puebla, Mexico, February 1997.

[4] "Applications of Data Mining for Power Systems," *Canadian Conference on Electrical and Computer Engineering*, St. John's, NF, May 1997, pp. 403-406.

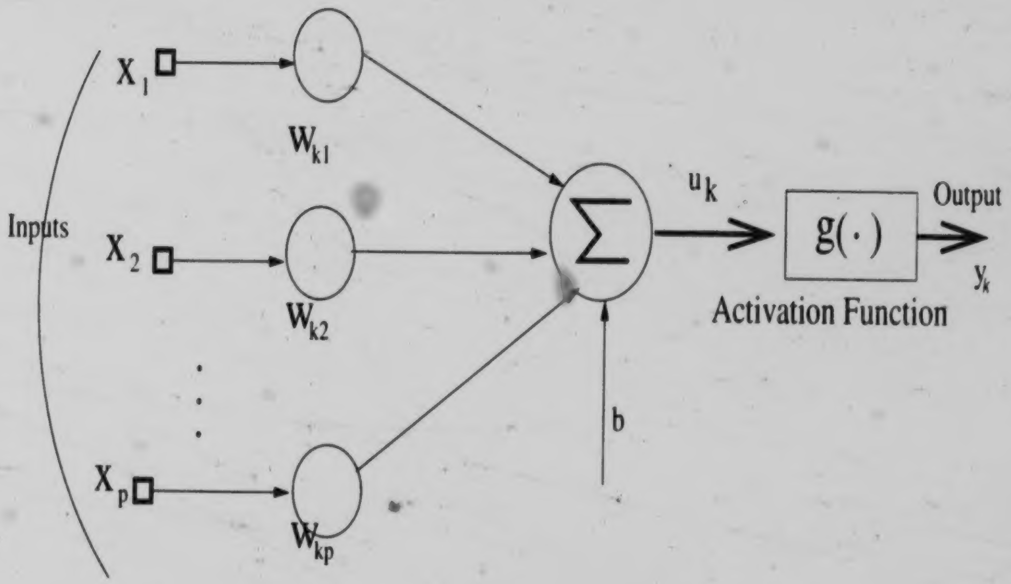


Figure 3.2: Model of a Neuron

There are three basic elements in the neuron model. They are described below:

1. A set of **synapses,** each of which is characterized by a *weight* or *strength* of its own. An input signal x_i at the synapse i connected to neuron k is multiplied by the synaptic weight w_{ki}. The weight w_{ki} is positive if the associated synapse is excitatory; it is negative if the synapse is inhibitory.
2. An **adder** for summing the input signals, weighted by the respective synapses of the neuron.
3. An **activation function** for limiting the amplitude of the output of a neuron. It is also referred to as a *squashing function.* The output range is normalized between the range [0, 1] or [-1, 1].

The model of the neuron as shown in Figure 3.2 includes a *bias* term, b, which serves to increase the net input to the activating function. Mathematically, the above neuron k may be described by the following pairs of equations:

$$u_k = \Sigma w_{ki} x_i \tag{3.7}$$

and

$$y_k = g(u_k + b) \tag{3.8}$$

where $x_1, x_2, \ldots x_p$ are the input signals, $w_{k1}, w_{k2}, \ldots w_{kp}$ are the synaptic weights of the neuron k, u_k is the linear combinator output, b is the bias, g is the activation function, and y_k is the output signal of the neuron.

3.4.2 Neural Network Architectures

The neural network architecture defines the ways in which the neurons are inter-connected (the structure). The classification [Haykin94] follows:

1. **Single-Layer Feedforward Networks:** A layered neural network is a network of neurons organized in the form of layers. In the simplest form of a layered network, there is an input layer of source nodes, that project onto an output layer of neurons, but with no response in the reverse direction. An example of a single-layered feedforward network with four nodes in both the input and output layers is shown in Figure 3.3. Here the designation *single layer* refers to the output layer of the computation nodes (neurons).

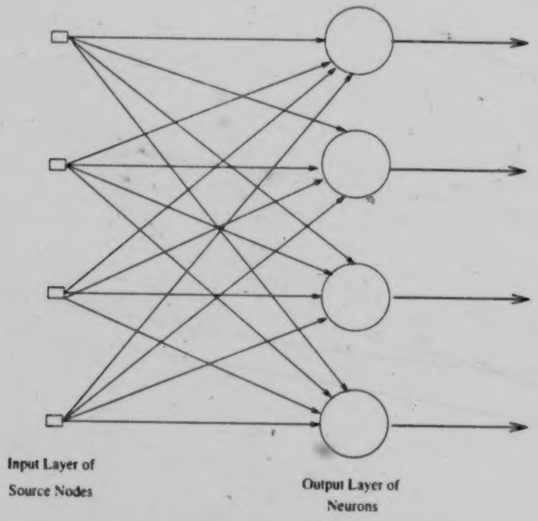


Figure 3.3: Single-layer Feedforward Networks

2. **Multilayer Feedforward Networks:** This class of networks is distinguished by the presence of one or more *hidden layers*; the computation nodes, called *hidden neurons* or *hidden units* are shown in Figure 3.4.

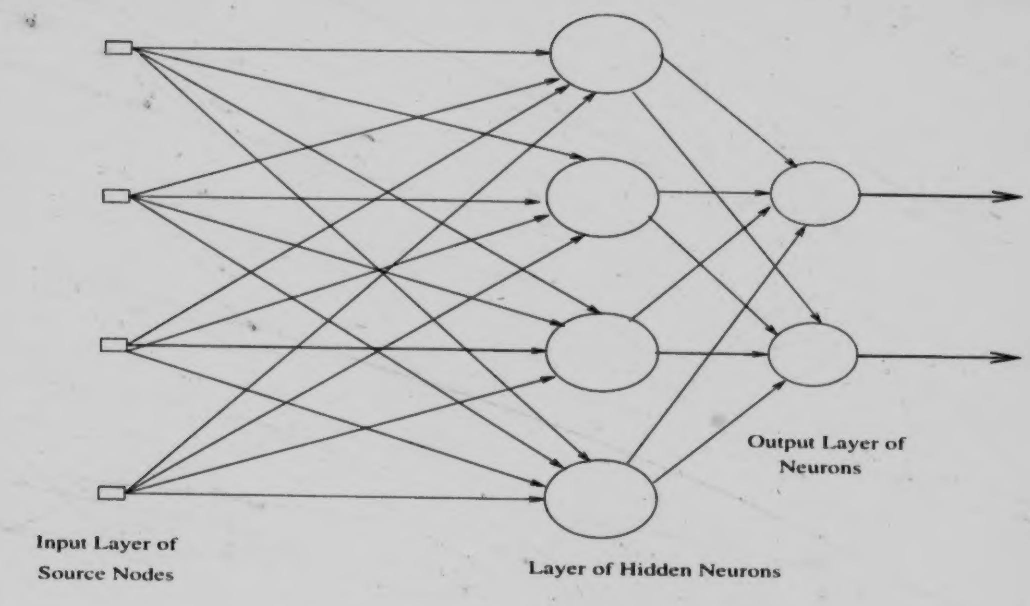

Input Layer of Source Nodes
Layer of Hidden Neurons
Output Layer of Neurons

Figure 3.4: Fully Connected Multilayer Feedforward Networks

The hidden neurons serve to intervene between the external input and the network output. By addition of one or more hidden layers, the network is able to extract input-output mapping relationship more accurately.

3. **Recurrent Networks:** A *recurrent neural network*, as shown in Figure 3.5, is different from a feedforward neural network in that it has at least one feedback loop.

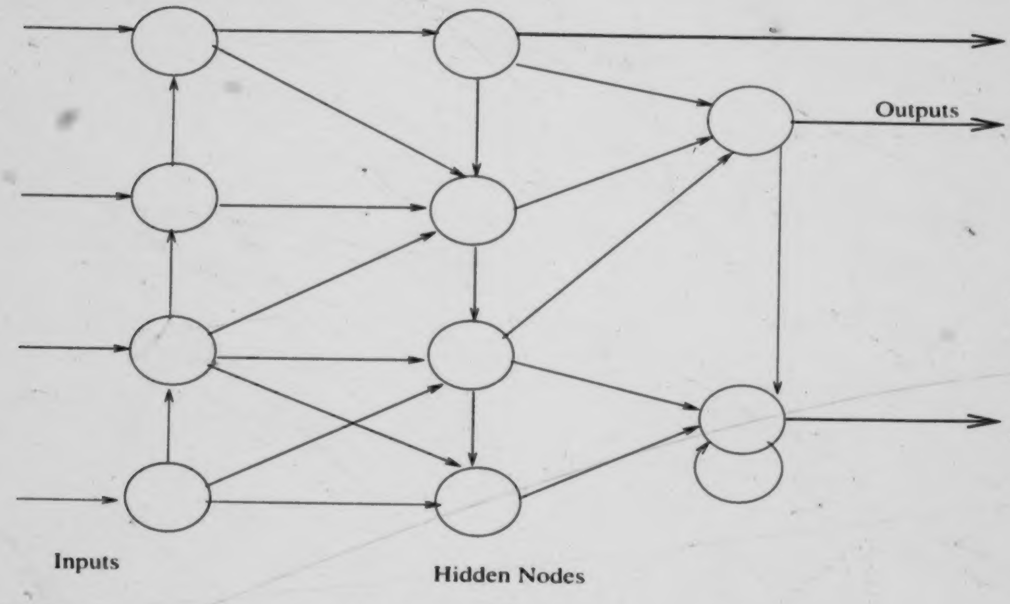


Figure 3.5: A General Recurrent Neural Network

4. **Lattice Structures:** A *lattice* consists of a one-dimensional, two-dimensional, or higher-dimensional array of neurons with a corresponding set of source nodes that supply the input signals to the array. The dimensions of the lattice refer to the number of the dimensions of the space in which the graph lies. A one-dimensional lattice of 3 neurons is shown in Figure 3.6.

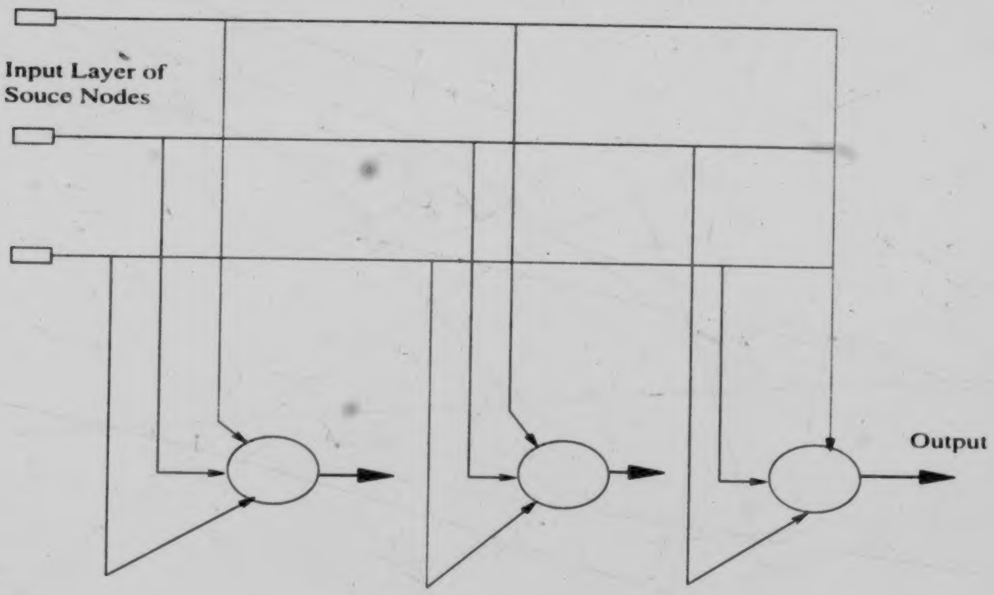


Figure 3.6: A One-dimensional Lattice of 3 Neurons

3.4.3 Comments

The advantages of artificial neural networks may be summarized as follows [Haykin94]:

1. Artificial neural networks have the ability to acquire complex, nonlinear relationships.
2. A properly trained neural network has a quick response.
3. They are adaptable. Neural networks have a built-in capability to adapt their synaptic weights to changes in the surrounding environment. Also, a neural network can be easily *retrained* to deal with small changes in the operating environmental conditions.

The limitations of artificial neural networks may be summarized as [Haykin94]:

1. They have problems associated with sensitivity. Neural networks are good at interpolation, but not at extrapolation. This means that the training set must comprise the complete pattern space that needs to be properly classified. This affects the sensitivity of artificial neural networks in providing accurate results. For example, if the training set includes a large range of numerical values, information provided by a small change of certain parameters can be lost.
2. There may be a lack of accuracy in the basic mathematical operations.

3. Their performance depends upon the topology and training. There are no exact guidelines as to the choice of the number of hidden layers, and to the number of neurons per hidden layer. The choice of the network topology and the procedures for selecting training sets depends strongly upon the actual problem.

To summarize, while ANNs have the problem of proper selection of architecture and training, once trained they can provide rapid predictions and classification on input data. In this way they can overcome the weakness of expert systems which do not have this ability. In this work, neural networks have been used to predict the new power flows over the transmission lines and transformers after a change in the setting of the controllers or load-shedding.

3.5 Conclusion to Chapter 3

This chapter has presented and summarized the main features of the enabling technologies that have been used for this research work: expert systems, artificial neural networks, and data mining. Each of these technologies has some limitations. For instance, expert systems have a problem in the knowledge acquisition stage. Further, they cannot predict values. Data mining is emerging as a viable tool for knowledge acquisition. Artificial neural networks have the ability to predict values. Considering the advantages and drawbacks of these methods, it appears that a more robust system can be developed if these techniques are used in a complementary fashion.

Chapter 4

AI and POWER SYSTEMS

This study investigates the development of an integrated hybrid intelligent assistant for power system operations. The developed system incorporates an expert system, a neural network and a data mining system, along with analytical techniques. This system provides guidance to operators for getting maximum performance from the power system. This chapter has the following objectives:

1. To present an overview of applications of artificial intelligence techniques to power systems.
2. To clarify the needs and benefits for integration of artificial intelligence techniques to power systems.
3. To examine the emerging power system scenario and problems that power system operators may encounter.
4. To describe the role that artificial intelligence techniques can play in mitigating some of these problems.

4.1 Power System Problems

Most of the time a modern interconnected electrical power system provides quality (a continuous and well-regulated) supply to the consumers. However, intermittently, the power system is subjected to disturbances that lead to interruptions of the power supply to customers. Power system planners try to design reliable systems that are able to cope with random disturbances and outages. But, even for a best planned system, unforeseen events can occur that stress the system beyond the planned limits. The reasons for this include:

1. A practically infinite number of possible operating modes and disturbances, especially in modern, extensively interconnected power systems.
2. Unpredictable structural changes due to the development of power systems that are subject to economical and environmental restrictions, and which consequently, generate dynamic changes.
3. A combination of unusual and undesired events (for example, human error combined with unfavorable weather) and/or planned maintenance of important system element(s).
4. The design philosophy of *planned reliability* is pushing power systems close to their operating limits, because this is economically desirable.

While reliability is the chief concern of system designers and planners, operators have to deal with system security, which is an on-line, operational characteristic of the power system. Operators have to cope with possible disturbances to ensure minimum service interruptions.

4.1.1 Causes

Disturbances manifest in a variety of ways and in a random manner. Thus, the progression of a disturbance is difficult to predict. A major disturbance starts with an initial event. The initial causes of a disturbance can be grouped into three basic categories:

1. equipment malfunction
2. human errors
3. natural events.

The events that contribute to further system degradation are[1]:

1. **overload of the system elements**
2. **voltage/reactive power problems**
3. **instability, including voltage instability**
4. protection coordination

---

[1] The bold items are areas of interest in this work

5. human errors in control actions (including false communication)

6. **system control deficiencies**

7. equipment malfunctioning.

Human error is a very significant cause in both the initial and the contributing events. The lack of appropriate communications (among operating personal) and inadequate information processing is a prominent contributing cause for further system degradation. The interconnections, interchange volume, and the complexity are rapidly becoming more important factors in a disturbance due to an interchange increase. The location of the power system and weather conditions strongly influence the vulnerability of the power system. For example, the western states of Canada and U.S.A. have been the scene of major disturbances over the last decade because of long distances involved. Some of these incidents have been documented in [IEEE93, Taylor94] and the URL site: *http://www.gridco.com/*.

4.1.2 Steps in Recovery

The existing tools in the EMS environment do not have facilities for the processing and detailed analysis of disturbances. The operator must deal with complex situations and rely on heuristic solutions. The steps involved in dealing with a problem are:

1. **Identification:** From the condition of the power system features (explained in the next section), immediate identification of an insecure situation is required to start the recovery procedure.

2. **Classification:** Disturbance classification is based on the severity of constraint violations and the combination of violations. The classification includes the recognition of the origin of the disturbance. For example, the procedure is different if an emergency is caused by an internal (e.g. over-voltage) or an external (e.g. line snapping) event.

3. **Choice of Corrective Measures:** The choice of corrective measures is directly related to the level of priority during the disturbance. The levels are:

   - stop the degradation of the system
   - return the system to a secure state
   - consider the economical and social impacts.

Often, to ensure satisfaction of priorities, suboptimal actions are performed.

4. **Coordination:** Different measures are used to solve the same problem, and uncoordinated actions can worsen the situation. Furthermore, different measures are used for different problems, but interactions can be strong. For example over-voltage may result when switching on capacitors to rectify under-voltage problems.

5. **Corrections:** After control measures are applied, the system can be in an improved but still in an unsatisfactory state. Additional corrective actions are required. Depending upon the priority level of the disturbance, applied action levels are usually not optimal. Once the system is in a secure state, optimization methods can be applied (for example partial restoring after load-shedding).

6. **Time:** For any of the previous steps, available time is a vital factor in selecting appropriate actions. The decision regarding the selection of controllers includes the time available for the choice of the method and implementation of the possible effective measures. A trade-off between optimal methods and time is very important.

4.1.3 Power System Features

The state of the power system is represented by the network features. Their appropriate identification is essential for any action during a disturbance. Potential features are summarized as follows:

1. **Active power flows in the network:** When the established limits are violated, the system is in a viability crisis. For the overloaded transformer and transmission lines, equipment life is shortened. Therefore, guidelines for transmission and transformer loadings are established to ensure long life. These guidelines influence the emergency procedure. The limits for the transmission line loading is set by transient and steady-state stability conditions (long lines), voltage collapse conditions (medium lines), and thermal conditions (short lines). The thermal capabilities for transmission elements are a matter of engineering judgment, and utilities define Short Term Emergency and Long Term Emergency ratings. Also, ambient temperature, loading history, and time constants of the overloaded element(s) need to be considered for accurate analysis.

2. **Voltage and reactive load flows:** When the established voltage limits of load buses and reactive power of generators are violated, the system is in a viability crisis. The voltages in the power system as well as sudden voltage changes need to be kept within

a small range. A voltage stability crisis has been one of the major reasons for system collapse in recent years. If reactive power support (required in local areas) is not provided, voltage drop can lead to voltage collapse. The values of the voltages and rate of voltage changes at certain buses can be used to identify and to classify the emergency.

3. **Stability** limits are indicated by the difference in angles between lines; these should be held within certain limits. Detection of an out-of-step condition can prevent instability, and consequently *cascading*. Cascading is a phenomenon which occurs when successive parts of the power system are cut off in succession.

4. **Frequency** deviation from the nominal value is a result of real power imbalance.

5. **Rate of Change of Frequency** is an instantaneous indicator of power deficiency in the islanded area.

6. **Spinning Reserve** can play a significant role within few seconds of a disturbance. The spinning reserve needs to be considered in load-shedding schemes to minimize the amount of load to be shed.

7. **Cold Reserve** influences the time required for restoring the system.

8. **Load Dependence of Voltage and Frequency:** Change of loads with voltage and frequency are important to avoid a transition to an unwanted state and/or to obtain optimum response to applied measures.

9. **Weather/Season:** The weather/seasonal changes directly influence system operation and security level, and thus the system response to a disturbance. The approach of a severe storm can transfer the system from a normal to an alert state.

10. **Configuration** determines the security level of the system.

4.1.4 Corrective Measures

The conditions of the above features help identify and classify an emergency. Different conditions demand different measures. Some important corrective measures[2] are:

- **Load-shedding**
- Controlled power system separation

---

[2] The bold items are areas of interest in this work

- Generation dropping
- Fault Clearing
- Fast Valving
- Dynamic braking
- **Generator Voltage Control**
- **Capacitor/Reactor Switching**
- **Load Control**
- **Voltage Reduction**
- **Transformer Tap Changing Action**
- Phase Shifting
- Tie Line Rescheduling
- Reserve Increasing
- Generator Shifting.

The values of the power system parameters and the sensitivity of the system to corrective measures are the factors that influence correction strategies. Certain measures can have adverse consequences that can provoke new problems, e.g. load-shedding which is one of the subject under study in this thesis.

Some adverse consequences of uncontrolled load-shedding include:

1. a possible undesired increase of the voltage (due to high shunt capacitances in the network),
2. an undesired increase in line flows, due to incorrect disposition of the load-shed, and
3. for over-shedding, the most dangerous consequence is tripping of the generator units due to an overfrequency condition, leading to blackout.

4.2 Power System Scenario

A modern interconnected power system is characterized by an increasing number of operating scenarios during a major disturbance. Unavoidably, the operation of the power system is considerably different from the vision of the system's designers. Current practices in power system planning lead power systems to work on tighter margins, with less redundancy, because of the difficulties in building new generating capacity (economic and social factors), and because of new demands. This leads to a decrease in the reliability of the power system which directly impacts the security level of the power system operation during a disturbance.

In the face of a disturbance, the operators are under the constraint of available time. Moreover, propagation of a major disturbance is difficult to incorporate into suitable numerical algorithms, and heuristics are required. Experienced and well-trained operators can recognize the situation and react properly, but sometimes not reliably and quickly enough. Along with the fact that the EMS is not designed for unexpected situations, automatic actions are conceived for a fast developing, serious disturbance. These fast actions are usually conservative, without taking into account the instantaneous state of the whole disturbed power system. Also, they are reactive as they do not anticipate the problem. Furthermore, possible consequences are not considered. For example, under-voltage relays operate when a voltage drops below a threshold based on local measurements [Taylor92].

Due to developments in computer hardware and software, communications, and control theory, a modern EMS in a control center is available to transmit and process data. It assists the operator to implement preventive actions efficiently in a normal or in an alert state. During an emergency, heuristic procedures and local protective relaying actions are still of prime importance because of the fast dynamic changes in the system parameters.

The propagation of a disturbance depends on the system response to an unplanned initial event. This response is greatly influenced by the specific features of the affected system. General mathematical methods cannot incorporate some of the above features (example weather and seasonal) and cannot take possible advantage of applying them. Further these methods may be too slow and probably require significant computer resources. The closer the method is to model the problem precisely, the more difficult it is to apply and understand it. This has led to consideration of schemes that consider heuristics.

4.3 Role of Operator

Automatic actions such as load-shedding are convenient and have the advantage of speed, simplicity, and reliability. With proper coordination of relays, low priority consumers may be disconnected, but a proper amount of load may not be shed. Further, load priorities may be different for the different seasons or even with the time of the day.

As noted above, managing an emergency is left to fast, reactive and non-adaptive automatic actions and/or to the slow and non-consistent operator. Experienced operators manage the situation by applying their experience, knowledge, and judgment. Combining these operator advantages with the elimination of operator weakness seems to be a good way to optimize control actions during a power system disturbance.

In order to avoid the drawbacks of conventional technologies, more sophisticated methods need to be applied. The objective of this research is to examine and apply technologies such as expert systems, neural networks, and data mining to improve methods of identifying major power system disturbances and to accelerate and optimize the available measures to prevent blackout.

For the problems under consideration (voltage/VAr violations and voltage stability) two types of decisions are required by the operators. They are:

1. Compute changes in controller settings so as to adjust the system parameters in such a way that the voltages at the load buses and the reactive powers of the generators are within the specified limits.
2. Compute the amount of load-shedding, when available resources are not able to maintain the system parameters within the limits and/or there is an imminent danger of voltage collapse.

In this thesis, these decisions are made by using AI techniques in conjunction with analytical techniques.

4.4 Role of Artificial Intelligence

Rapid, adaptive, and proactive decisions are required in applying measures like load-shedding or changes in controller settings in order to relieve voltage/VAr violations. These system violations need to be alleviated within a time period that depends upon the overload severity, weather, temperature, and loading history. If violations (such as under-voltage, over-voltage) are not alleviated within a certain time, a process of power system cascading starts, which leads to the formation of isolated areas (islands) that have an imbalance between power generation and load. If corrective actions, which are usually relatively slow, are not sufficient or are not activated, the most common outcome is power system voltage collapse. These automatic measures are not immune to over-shedding and under-shedding because of their reactive nature.

With current technologies, the above requirements for controller settings and load-shedding are not fulfilled to a satisfactory degree, which leads to the strategies suggested in this thesis. Presently these decisions are taken based on operator experience, browsing operational manuals, and largely ad-hoc approaches.

Expert systems have proved to be helpful for power system operations, particularly in the presence of numerous data, by providing diagnosis, analyzing, selecting, predicting, and instruction capabilities to operators [Madan97b][3]. However expert systems are neither adaptive nor able to cope with unfamiliar situations. Artificial neural networks have the ability to acquire internal nonlinear relationships among power system variables to perform required pattern recognition for a disturbance. The weakness of expert systems in number crunching can be avoided by developing hybrid systems (for example, by combining ANN with ES). This new intelligent system can make rapid, adaptive, and proactive decisions which are required in applying measures like load-shedding or changes in controller settings.

The hybrid system provides certain advantages compared to existing technologies. An exhaustive analysis is performed in the off-line mode and embedded in the system: when needed this can furnish rapid and intelligent decisions. A combination of an expert system with a neural network seems to be a promising attempt to optimally solve the problems that arise during an emergency. The advantages of neural networks are adaptability and speed, which, combined with expert system capabilities, offer an adequate choice for emergency control.

---

[3] "Applications of Artificial Intelligence for Power Systems," *Electric Power Systems Research Journal*, vol. 41, no. 2, May 1997, pp. 117-131.

Heuristic approaches and training sets are generated to develop an intelligent system for the specific power system. This knowledge is also referred to as the *private knowledge* of that specific power system.

4.5 Conclusion to Chapter 4

This chapter explained the role of artificial intelligence for power system operation. Causes of power system problems along with corrective measures were reviewed. This chapter also explained why existing Energy Management Systems in a modern interconnected power system fall short of the need of the operators. This has changed with the advent of artificial intelligence technologies. Emerging modern technologies that model human behavior such as expert systems and neural networks offer a viable solution to power system problems. This includes the problem of voltage/VAr violation and voltage stability. An expert system model allows incorporation of qualitative and heuristic knowledge along with analytical techniques. The training sets presented to the neural networks describe the state of the system during various scenarios which can be used to make predictions for the new state. An hybrid approach that combines the best features of each technology seems to be most promising.

In this thesis, an integrated hybrid intelligent system has been developed for voltage/VAr violations and voltage stability. This has been achieved by combining AI based and analytical techniques in a novel way. The next chapter describes the models and algorithms developed in this thesis.

Chapter 5

MODELS and ALGORITHMS

This chapter describes power system models and algorithms developed for studying power systems functioning under normal and disturbed conditions. These models and algorithms enable an analysis of the functioning of the power systems, formulation of correction strategies, and provide a means to verify that these strategies will be effective. The following models and algorithms are described:

1. The **V**oltage reactive power **Q** **S**ensitivity **M**ethod (VQSM) determine the voltage stability of the system, based on the reduced Jacobian matrix [Kundur94].
2. The **V**oltage **V**Ar **C**ontroller **M**ethod (VVCM) recommends new settings for the controllers, that would remedy any voltage/VAr violations. The VVCM incorporates an expert system based monitoring system called the **E**xpert **S**ystem **M**onitor (ESM). The ESM determines the presence of voltage/VAr violations and also indicates the severity of the problem. The algorithm for the VVCM is outlined and incorporated production rules are also described.
3. The VVCM operation is based on the sensitivity of the controlled (dependent) variables w.r.t. the controller variables. The computation of the sensitivity matrix, *[S]* is described. The sensitivity matrix describes the relationship between changes in the controller variables (tap setting of transformers, voltage of generator buses, switching of capacitors/reactors at load buses) and their effect on the controlled variables (reactive power of generators and voltages at load buses).
4. To supplement the VVCM, methods based on *linear programming* and *mixed integer linear programming* to solve voltage/VAr violations have been formulated.
5. The **Load Shedding Methods** determine the location and amount of loads to be shed. Three methods have been described. These methods also predict the effect of load-

shedding on the rest of the system.

6. An algorithm for generating the *database* for training a neural network, and for inducing a decision tree is described. The neural network is used to predict the new power flow over the transmission lines and transformers once the settings of controllers have been changed or load-shedding has been done. The decision tree classifies the power system operating mode into *stable*, *intermediate stable* or *unstable* mode depending upon power system parameters.

New models and algorithms, and improvements that have been made for existing models and algorithms, have been identified. For example, an improved model pertaining to the sensitivities of the reactive power of generators and load bus voltages w.r.t. change in transformer taps has been presented. This model, when compared with similar models [Singh93, Keyhani89] as reported in the literature gives better results. Two new models, which have not been previously reported in the literature, have been presented for load-shedding.

The end product, which incorporates all the above methods, is called the **C**omprehensive **V**oltage **V**Ar **C**ontroller (CVVC) toolbox. A summary of the programs developed in this research work has been given in Appendix 'B'.

5.1 Methodology

The overall methodology used by the **C**omprehensive **V**oltage and **V**Ar **C**ontroller (CVVC) toolbox is described below [Madan97e][1]. For the following discussion, reference may be made to Figure 5.1.

---

[1] "A Hybrid Intelligent System for Voltage and VAr Problems in Power Systems," *29th North American Power Symposium*, Laramie, Wyoming, October 1997, pp. 76-83.

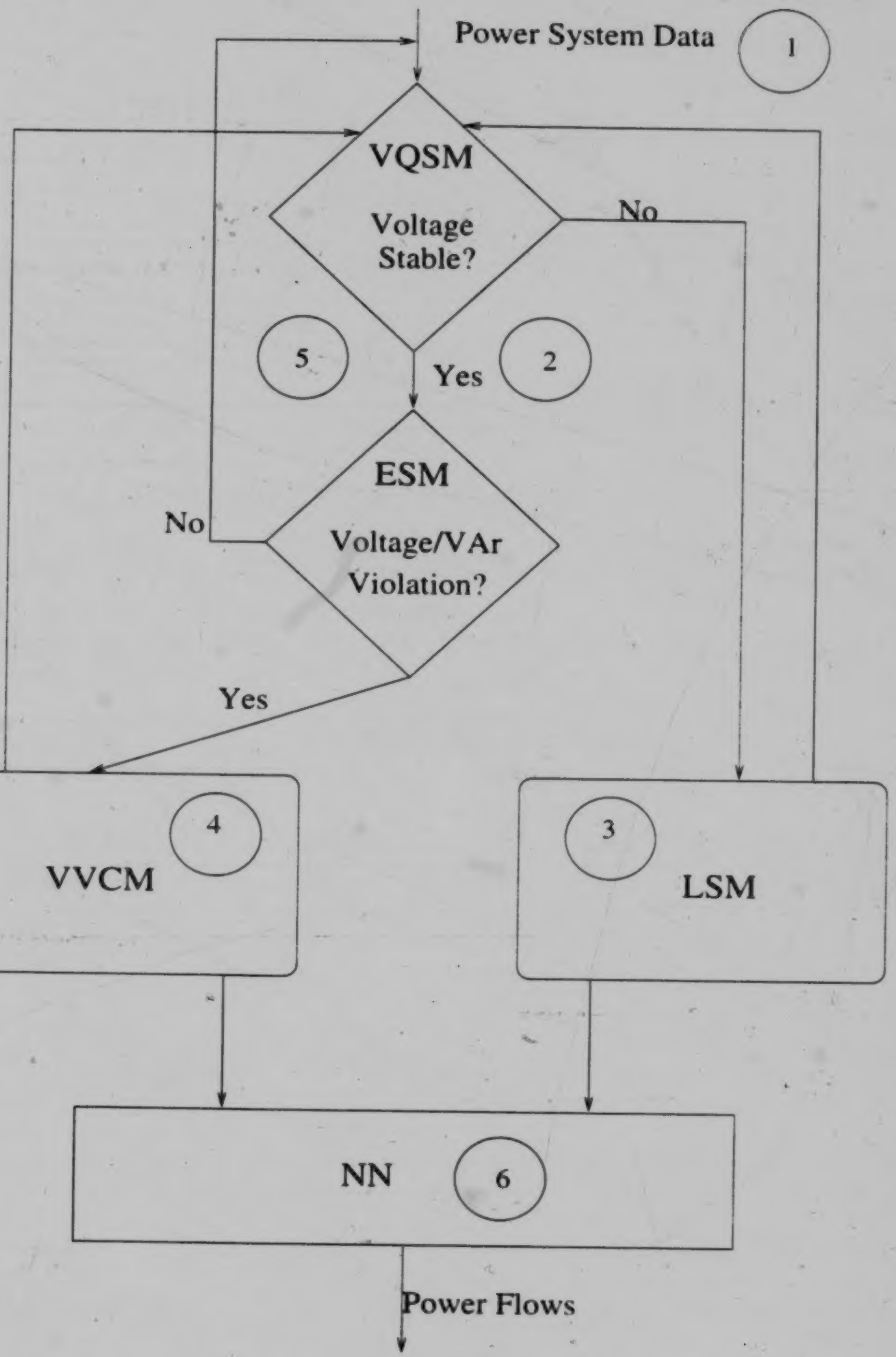

Power System Data
1
VQSM
Voltage Stable?
No
5
Yes
2
ESM
No
Voltage/VAr Violation?
Yes
4
VVCM
3
LSM
NN
6
Power Flows

Figure 5.1: Structure of CVVC Toolbox

1. **Step 1:** The load flow is run (or alternatively the power system data is telemetered in). The sensitivity matrix is formulated by using the topology of the connected network (admittance matrix) and system parameters (generator and load bus voltages).

2. **Step 2:** A check is made whether the system is voltage stable or not? If the system is voltage stable, then a check is made for voltage/VAr violation(s). If the system is normal (voltage stable and no voltage/VAr violations), then the program reports the status of the power system as normal and the program control returns to the previous step. If the system is voltage unstable the program control goes to step 3 (this is referred to as the Load-Shedding Method (LSM)). If the system is voltage stable but has voltage/VAr violation(s), then the program control goes to step 4 (this is referred to as the Voltage VAr Controller Method (CVVC)). It may be noted that a check is made first for voltage stability, since voltage stability problems are more serious than voltage/VAr violations.

3. **Step 3:** This step takes care of voltage instability by resorting to load-shedding. The **Load Shedding Method** (LSM) suggests the amounts and location of the loads to be shed. This module also reports the effect of the suggested measures. Then the program control goes to step 5.

4. **Step 4:** This step takes care of voltage/VAr violations. The Voltage VAr **Controller** Method (VVCM) makes a choice of controller settings to correct the problem. This module also reports the effect of the suggested measures. Then the program control goes to step 5.

5. **Step 5:** A check is made whether the system is normal (voltage stable and no voltage/VAr violations). If the system is normal, then the program control goes to step 1. If the system has developed voltage stability problems or if severe voltage/VAr violations persist, then the CVVC toolbox recommends load-shedding and goes to step 3. ***Step 5 is the same as step 2.***

6. **Step 6:** In this step the CVVC toolbox invokes the neural network based system to predict power flow levels over the transmission lines and transformers.

5.2 Formulation of VQSM

This formulation is derived from [Kundur94]. A converged load flow solution based on Newton Raphson's method has the following valid relations:

$$\begin{bmatrix}\Delta P\\ \Delta Q\end{bmatrix} = \begin{bmatrix}J_{P\theta} & J_{PV}\\ J_{Q\theta} & J_{QV}\end{bmatrix}\begin{bmatrix}\Delta\theta\\ \Delta V\end{bmatrix} \tag{5.1}$$

or

$$\begin{bmatrix}\Delta P\\ \Delta Q\end{bmatrix} = [J]\begin{bmatrix}\Delta\theta\\ \Delta V\end{bmatrix} \tag{5.2}$$

where

ΔP = incremental change in bus real power injection

ΔQ = incremental change in bus reactive power injection

$\Delta\theta$ = incremental change in bus voltage angle

ΔV = incremental change in bus voltage magnitude

The [J] matrix is the Jacobian matrix which gives the sensitivity between power flows and bus voltage changes.
If $\Delta P = 0$, then

$$\Delta Q = J_R \Delta V \tag{5.3}$$

where

$$J_R = J_{QV} - J_{Q\theta} J_{P\theta}^{-1} J_{PV} \tag{5.4}$$

is the reduced Jacobian matrix of the system. Therefore:

$$\Delta V = J_R^{-1} \Delta Q \tag{5.5}$$

The *i th* diagonal element of J_R is the V-Q sensitivity at bus *i*. The V-Q sensitivity at a bus represents the slope of the Q-V curve at a given operating point. A positive V-Q sensitivity is indicative of stable operation; the smaller the sensitivity, the more stable the system. As the stability decreases, the magnitude of the sensitivity increases, becoming infinite at the stability limit. Conversely, a negative V-Q sensitivity is indicative of unstable operation.

5.3 Expert System Monitor (ESM)

This section explains the development of a voltage and VAr monitoring system which has been developed using CLIPS, an expert system shell. Monitoring problems tend to be well-suited for forward chaining rule-based languages because of their data-driven nature. In the setup, a set of input or sensor values are read during each program cycle. Inferencing occurs until all possible conclusions which can be derived from input data are reached. This is consistent with a data-driven approach in which reasoning occurs from data to conclusions that can be derived from the data.

A power system contains several devices which have to be monitored. For the problem under consideration this involves monitoring of reactive power of generators and voltages at load buses. It is assumed that each device has one sensor attached to it which provides numeric readings indicating the health of the device. Each sensor has an average expected value (AEV) associated with it, as well as low-guard line (LGL), low red line (LRL), high-guard line (HGL), and high red line (HRL) values. A reading between the low and and high-guard line is considered as normal. A reading above the high-guard line but below the high red line, or below the low-guard line but above the low red line is considered acceptable, although it is an indication that the device may soon become unhealthy. A reading above the high red line or below the low red line indicates an unhealthy device.

The developed monitoring program reads in sensor data, evaluates the sensor readings, determines trends, issues warnings, and gives recommendations based upon sensor evaluations and trends.

Some of the assumptions are:

1. Sensor data is always available upon request and is always reliable.
2. Increasing and decreasing trends are considered as an increase or decrease of a specified value over a specified period of time.

The problem is split into three phases; reading of sensor values, analysis of sensor values, and appropriate actions. They are:

1. **Input Phase:** The first phase is the user supplied phase or the input phase.
2. **Detecting a Trend;** The second phase associates guardline and redline conditions for the sensor values and determines any developing trends. The current state of the sensor (normal, low or high guard line, or low or high red line) is determined from the raw sensor value asserted during the input phase.

3. **Issuing Warnings:** The final phase of the cycle is the warning phase. Three types of actions are handled during this phase: sensors that have entered a red line region, are classified under the *in-extremis* state, sensors that have stayed in the guard line region for a specified number of cycles are classified under the *alert* state, and sensors that are in the guard line region issue a *warning*.

5.4 Formulation of Sensitivity Matrix

The second equation from the fast de-coupled load flow [Stott74] equations may be written as:

$$[\Delta Q/V] = [B''][\Delta V] \tag{5.6}$$

where

ΔQ is incremental change in reactive power

V is the voltage

$-B''$ is the imaginary part of bus admittance matrix

ΔV is incremental change in voltage.

Buses are numbered so that the first set of buses represent the generator (PV) buses and then the load (PQ) buses. Equation 5.6 is extended to show the relationship between the generator and load buses,

$$\begin{bmatrix} \Delta Q_g/V_g \\ \Delta Q_l/V_l \end{bmatrix} = \begin{bmatrix} B_{gg} & B_{gl} \\ B_{lg} & B_{ll} \end{bmatrix} \begin{bmatrix} \Delta V_g \\ \Delta V_l \end{bmatrix} \tag{5.7}$$

where subscripts g and l denote the PV and the PQ buses respectively.

Using equation 5.7, one can write a sensitivity matrix showing a relationship between changes in the controller elements and the corresponding effect on the generator and load buses. Considering t number of transformers with taps, n number of generator and m number of load buses, one can write:

$$\begin{bmatrix} \Delta Q_{g1} \\ \cdots \\ \Delta Q_{gn} \\ \Delta V_{l1} \\ \cdots \\ \Delta V_{lm} \end{bmatrix} = \begin{bmatrix} S_{Q_gT} & S_{Q_gV_g} & S_{Q_gQl} \\ S_{VlT} & S_{VlV_g} & S_{VlQl} \end{bmatrix} \begin{bmatrix} \Delta T_1 \\ \cdots \\ \Delta T_k \\ \Delta V_{g1} \\ \cdots \\ \Delta V_{gn} \\ \Delta Q_{l1} \\ \cdots \\ \Delta Q_{lm} \end{bmatrix} \tag{5.8}$$

The above equation may be abbreviated as:

$$[A] = [S][X] \tag{5.9}$$

where the [S] matrix is the sensitivity matrix. The [S] matrix has 6 sub-matrices which are calculated from equation 5.7. These sub-matrices are defined as follows:

S_{Q_gT} is defined as the change in the reactive power of the generators per unit change in the transformer tap settings.

S_{V_lT} is defined as the change in the load bus voltages per unit change in the transformer tap settings.

$S_{Q_gV_g}$ is defined as the change in the reactive power of the generators per unit change in the generator bus voltages.

$S_{V_lV_g}$ is defined as the change in load bus voltages per unit change in the generator bus voltages.

$S_{Q_gQ_l}$ is defined as the change in the reactive power of the generators per unit change in the reactive power due to the switching on/off the capacitors/reactors.

$S_{V_lQ_l}$ is is defined as the change in the load bus voltages per unit change in reactive power due to the switching on/off the reactors/capacitors.

Note: *The formulation of equation 5.7 has been derived from [Keyhani89, Singh93]. In this thesis, a modification was made by using complex impedance instead of simply the imaginary part of the bus admittance matrix. This gives more accurate sensitivity values, particularly for cases when the transmission line X/R ratio is not high.*

5.4.1 Sensitivity due to Generator Bus Voltage Change

This formulation has been derived from [Keyhani89, Singh93]. A change in voltage at any generator bus in an interconnected system will affect the reactive powers of the generator buses only. The reactive power of the load bus voltages will not change as long as the load remains constant. However, voltages of load buses will change. These effects can be accommodated by equating ΔQ_l to zero in equation 5.7 with the resulting expression:

$$\begin{bmatrix} \Delta Q_g/V_g \\ 0 \end{bmatrix} = \begin{bmatrix} B_{gg} & B_{gl} \\ B_{lg} & B_{ll} \end{bmatrix} \begin{bmatrix} \Delta V_g \\ \Delta V_l \end{bmatrix} \tag{5.10}$$

Expanding the above equation, one gets:

$$\Delta Q_g/V_g = B_{gg}\Delta V_g + B_{gl}\Delta V_l \tag{5.11}$$

and

$$B_{lg}\Delta V_g + B_{ll}\Delta V_l = 0 \tag{5.12}$$

Equation 5.12 can be re-written as:

$$\Delta V_l = -B_{ll}^{-1} B_{lg}\Delta V_g \tag{5.13}$$

The sensitivity of the load bus voltages w.r.t. the generator bus voltages is defined as the change in load bus voltages per unit change in the generator bus voltages:

$$[\Delta V_l] = [S_{V_l V_g}][\Delta V_g] \tag{5.14}$$

Comparing equations 5.13 and 5.14 :

$$[S_{V_l V_g}] = -[B_{ll}]^{-1}[B_{lg}] \tag{5.15}$$

Substituting the value of ΔV_l from equation 5.13 into equation 5.11.

$$\Delta Q_g/V_g = [B_{gg} - B_{gl}B_{ll}^{-1}B_{lg}][\Delta V_g] \tag{5.16}$$

or

$$\Delta Q_g = (Diag.[V_g])[B_{gg} - B_{gl}B_{ll}^{-1}B_{lg}][\Delta V_g] \tag{5.17}$$

The sensitivity of the reactive power of the generators w.r.t. the generator bus voltages is defined as the change in the reactive power of the generators per unit change in the generator bus voltages:

$$[\Delta Q_g] = [S_{Q_g V_g}][\Delta V_g] \tag{5.18}$$

Comparing equations 5.17 and 5.18:

$$[S_{Q_g V_g}] = (Diag.[V_g])[B_{gg} - B_{gl}B_{ll}^{-1}B_{lg}] \tag{5.19}$$

5.4.2 Sensitivity due to Capacitor/Reactor switching

This formulation has been derived from [Keyhani89, Singh93]. When the capacitor/reactor switching occurs, the change in setting will change the load bus voltages but the generator bus voltages will not change. However, the generator bus reactive generations will be affected.

For this case, therefore, equation 5.7 becomes:

$$\begin{bmatrix} \Delta Q_g/V_g \\ \Delta Q_l/V_l \end{bmatrix} = \begin{bmatrix} B_{gg} & B_{gl} \\ B_{lg} & B_{ll} \end{bmatrix} \begin{bmatrix} 0 \\ \Delta V_l \end{bmatrix} \tag{5.20}$$

Expanding the above equation:

$$\Delta Q_g/V_g = B_{gl}\Delta V_l \tag{5.21}$$

and

$$\Delta Q_l/V_l = B_{ll}\Delta V_l \tag{5.22}$$

or

$$\Delta V_l = B_{ll}^{-1}\Delta Q_l/V_l \tag{5.23}$$

or

$$[\Delta V_l] = [B_{ll}]^{-1}(Diag.[V_l]^{-1})[\Delta Q_l] \tag{5.24}$$

The sensitivity of the load bus voltages w.r.t. switching on/off the reactors/capacitors is defined as the change in the load bus voltages per unit change in reactive power due to the switching on/off the reactors/capacitors:

$$[\Delta V_l] = [S_{V_lQ_l}][\Delta Q_l] \tag{5.25}$$

Comparing equations 5.24 and 5.25:

$$[S_{V_lQ_l}] = [B_{ll}]^{-1}(Diag.[V_l]^{-1}) \tag{5.26}$$

From equation 5.21

$$\Delta Q_g = (Diag.[V_g])[B_{gl}][\Delta V_l] \tag{5.27}$$

Substituting the value of ΔV_l from equation 5.23 into the above equation

$$\Delta Q_g = (Diag.(V_g])[B_{gl}][B_{ll}]^{-1}[\Delta Q_l/V_l] \qquad (5.28)$$

or

$$\Delta Q_g = (Diag.[V_g])[B_{gl}][B_{ll}]^{-1}(Diag.[V_l]^{-1})[\Delta Q_l] \qquad (5.29)$$

The sensitivity of the reactive power of the generators w.r.t. switching on/off the capacitors/reactors is defined as the change in the reactive power of the generators per unit change in the reactive power due to the switching on/off the capacitors/reactors:

$$[\Delta Q_g] = [S_{Q_gQ_l}][\Delta Q_l] \qquad (5.30)$$

Comparing equations 5.29 and 5.30:

$$[S_{Q_gQ_l}] = (Diag.[V_g])[B_{gl}][B_{ll}]^{-1}(Diag.[V_l]^{-1}) \qquad (5.31)$$

5.4.3 Sensitivity due to Tap Change in Transformers

In this sub-section transformer models which have been presented in the literature are reviewed. In the following sub-section, a new transformer model developed in this research work is presented.

Reference [Singh93] gives the following sensitivity formulas without any details. A transformer of admittance *Y* and tap setting *1:k* is assumed to be connected between buses. *i* and *j*.

The sensitivity of the load bus voltages w.r.t. the transformer tap settings is defined as the change in the load bus voltages per unit change in the transformer tap settings:

$$[\Delta V_l] = [S_{VlT}][\Delta k] \qquad (5.32)$$

$$[S_{VlT}] = \begin{bmatrix} V_i(-2/k^3)Y + V_jY/k^2 \\ 0 \\ . \\ . \\ V_iY/k^2 \\ 0 \\ . \\ . \end{bmatrix} \qquad (5.33)$$

The sensitivity of the reactive power of the generators w.r.t. change in the transformer tap settings is defined as the change in the reactive power of the generators due to per unit change in the transformer tap settings:

$$[\Delta Q_g] = [S_{QgT}][\Delta k] \tag{5.34}$$

$$[S_{QgT}] = FACTOR * \begin{bmatrix} V_i(-2/k^3)Y + V_jY/k^2 \\ 0 \\ . \\ . \\ V_jY/k^2 \\ 0 \\ . \\ . \end{bmatrix} \tag{5.35}$$

where

$$FACTOR = (Diag.[V_g])[B_{gl}][B_{ll}]^{-1} \tag{5.36}$$

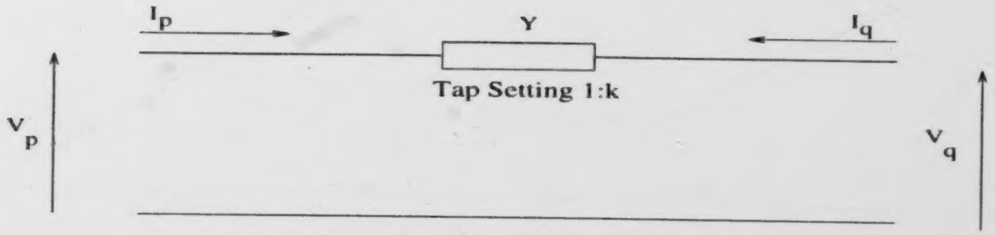
Reference [Keyhani89] gives some details of the above formulation. These details are reproduced below:

An ideal transformer of admittance Y is assumed to be connected between buses p and q as shown in Figure 5.2. The following relationships have been derived from Figure 5.2(b).

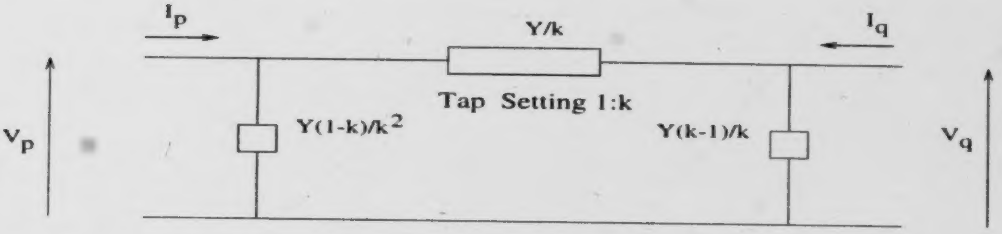
CANADIAN THESES
THESES CANADIENNES

MADAN, SANJAY
DEVELOPMENT OF AN INTEGRATED
HYBRID INTELLIGENT SYSTEM FOR ...

ISBN 0-612-29068-
NQ-2906
2 of



(a) Ideal Transformer Model



(b) Ideal Transformer Equivalent

Figure 5.2: Transformer Model

$$I_p = (V_p - kV_q)Y1/k^2 \tag{5.37}$$

$$I_q = (kV_q - V_p)Y1/k \tag{5.38}$$

Differentiating I_p and I_q w.r.t. change in transformer tap settings:

$$\Delta I_p = (V_pY(-2/k^3) + V_qY(1/k^2))\Delta k \tag{5.39}$$

$$\Delta I_q = (V_pY(1/k^2))\Delta k \tag{5.40}$$

For each tap changing transformer, the corresponding columns of S_{VIT} (defined by equation 5.32) and S_{QgT} (defined by equation 5.34) matrices are to be calculated. Assuming that a tap changing transformer is connected to the first two *internal buses* and using the concept of a *local network*[2], the reactive power change is:

---

[2] A local network consists of a group of internal buses, boundary buses and connecting lines. The load (P-Q) and inactive generator (P-V) buses are the internal buses. Active P-V buses and the slack bus are the boundary buses. In effect a local network is a portion of the bulk power system that has its boundary cut at the active buses. A voltage violation within the local network does not affect any facilities outside the network and a boundary bus control action only affects the adjacent local networks.

$$\Delta Q_i = [V_p \Delta I_p \quad \Delta V_q I_q \quad 0] \tag{5.41}$$

The following equation which relates changes in load bus voltages with reactive power changes at the load buses is the same as equation 5.23:

$$\Delta V_i = [B_{ll}^{-1}] \begin{bmatrix} \Delta I_p \\ \Delta I_q \\ 0 \end{bmatrix} \tag{5.42}$$

One notes that in equation 5.23, the right hand side has ΔQ_l / V_l which is equal to the change in current as seen in the above equation.

Substituting the value of ΔI_p and ΔI_q from equations 5.39 and 5.40 respectively in the above equation:

$$\Delta V_i = [B_{ll}^{-1}] \begin{bmatrix} V_p Y(-2/k^3) + V_q Y(1/k^2) \\ V_p Y(1/k^2) \\ 0 \end{bmatrix} \Delta k \tag{5.43}$$

Therefore, $[S_{V lT}] = [\Delta V_l][\Delta k]^{-1}$ becomes:

$$[S_{VlT}] = B_{ll}^{-1} \begin{bmatrix} V_p Y(-2/k^3) + V_q Y(1/k^2) \\ V_p Y(1/k^2) \\ 0 \end{bmatrix} \tag{5.44}$$

and $S_{QgT} = [\Delta Q_g][\Delta k]^{-1}$ is given by the following expression: (*without derivation*)

$$[S_{QgT}] = \begin{bmatrix} V_p Y(-2/k^3) + V_q Y(1/k^2) \\ (Diag.V_g)(B_{gl} B_{ll}^{-1} V_q Y) 1/k^2 \\ 0 \end{bmatrix} \tag{5.45}$$

5.4.4 Sensitivity due to Tap Change in Transformers - New Model

This sub-section presents a new transformer model for tap changing transformers, which has been developed as a part of this research. This model gives better results than those presented by [Keyhani89, Singh93]. The steps involved in computing the sensitivity of reactive power of generators and voltage of load buses w.r.t. to change in transformer tap setting are given below. A transformer of admittance (Y_t) is assumed to be connected between i and j buses in a system of size NBUS, whose admittance is denoted by YBUS.

1. **Step 1:** Identify the buses between which the transformers are connected.
2. **Step 2:** Compute the change in the admittance matrix (Y_t(change)) due to change in transformer tap setting (say from *1:y* to *1:z*):

$$Y_t(change)(i, i) = Y_t(i, i) - Y_t(i, i)$$
$$Y_t(change)(i, j) = (-Y_t(i, j)/y) - (-Y_t(i, j)/z)$$
$$Y_t(change)(j, i) = (-Y_t(j, i)/y) - (-Y_t(j, i)/z)$$
$$Y_t(change)(j, j) = Y_t(j, j)/(y^2) - Y_t(j, j)/(z^2)$$

3. **Step 3:** The change in transformer tap ratios[3] changes the admittance matrix for that part of the system in which this transformer is connected (between buses *i* and *j*):

   $$YBUS(new)(i, i) = YBUS(old)(i, i) + Y_t(change)(i, i)$$
   $$YBUS(new)(i, j) = YBUS(old)(i, j) + Y_t(change)(i, j)$$
   $$YBUS(new)(j, i) = YBUS(old)(j, i) + Y_t(change)(j, i)$$
   $$YBUS(new)(j, j) = YBUS(old)(j, j) + Y_t(change)(j, j)$$

4. **Step 4:** For computing load bus voltage changes with change in transformer tap settings one has to take into account the type of buses (generator or load) to which the transformer ends are connected. A change in current ΔI because of the transformer tap change is computed. An algorithm to compute ΔI is given at the end of this sub-section.

   The change in transformer tap ratio is:

   $$\Delta k = y - z \qquad (5.46)$$

   The voltage increment $[\Delta V_l]$ at the load buses because of the transformer tap ratio change, is given by the following equation which follows from equations 5.42 and 5.43:

   $$[\Delta V_l] = [B_{ll}^{-1}] * [\Delta I] * [\Delta k] \qquad (5.47)$$

5. **Step 5:** The voltage sensitivity of the load bus voltages with respect to a per unit change in the transformer tap ratio is defined as:

   $$[S_{VlT}] = [\Delta V_l][\Delta k]^{-1} \qquad (5.48)$$

   Therefore, from equation 5.47:

   $$[\mathbf{S_{VlT}}] = [\mathbf{B_{ll}}]^{-1}[\mathbf{\Delta I}] \qquad (5.49)$$

---

[3] or transformer tap settings

The above description is for one transformer. *Steps 1 to 5 are repeated. for each transformer.*

The following steps are needed to compute S_{QgT}, the sensitivity of the reactive power of the generators due to a per unit change in the transformer tap ratio.

1. **Step 1:** For one transformer at a time, compute new load bus voltages using $[S_{VIT}]$ computed above. Then compute the new output reactive powers of the generators $[Q_g(new)]$ using the new admittance matrix $(YBUS_{new})$ of the system.

$$Q_g(new) = Imaginary(V_i^* \sum_{j=1}^{NBUS} (YBUS(new)_{ij} V_j)) \tag{5.50}$$

2. **Step 2:** Compute the differences in the reactive powers of the generators = (new reactive power of generators - old reactive power of the generators) for a Δt change in transformer tap ratio:

$$[\Delta Q_g] = [Q_g(new)] - [Q_g(old)] \tag{5.51}$$

Also:

$$[\Delta Q_g] = [S_{QgT}][\Delta k] \tag{5.52}$$

From the above equation, compute $[S_{Qgt}]$:

$$[\mathbf{S_{QgT}}] = (\mathbf{Diag}.[\mathbf{\Delta k}]^{-1})[\mathbf{\Delta Q_g}] \tag{5.53}$$

An Algorithm for Computing ΔI vector

The following algorithm is needed to compute S_{Vlt}, the sensitivity of the voltage at load buses w.r.t. a change in tap ratio of the transformers.

Defining: fac2 = $(tap\_ratio(old))^2$ and

fac3 = $(tap\_ratio(old))^3$;

In the following algorithm, the *bus* vector encodes the information about the type of buses: the *start* and *finish* buses identifies the serial number of buses between which the transformer is connected.

```
for k = 1:NBUS
    $\Delta I(k)$ =0
for l = 1:NBUS
    if (k == i & l == j)
        if (bus(start) == 'generator' & bus(finish) == 'load)
            $\Delta I(start) = 0$
            $\Delta I(finish) = Y_{old}(i, j)^*$ (volt(i)/fac2 - 2*volt(j)/fac3)
        elseif bus(start) == 'load' & bus(finish) == 'load')
            $\Delta I(start) = Y_{old}(i, j)^*$ (volt(j)/fac2)
            $\Delta I(finish) = Y_{old}(i, j)^*$ (volt(i)/fac2 - 2*volt(j)/fac3)
        else (bus(start) == 'load' & bus(finish) == 'generator')
            $\Delta I(start) = Y_{old}(i, j)^*$(volt(j)/fac2)
            $\Delta I(finish) = 0$
        end
    else
        {do nothing}
    end
end
end
```

5.5 The VVCM Algorithm

1. The system parameters (transformer tap ratios, voltages at all the buses, load and generated real and reactive powers) along with the topology of the connected power

system network are read or telemetered.

2. The limits of the system parameters are read. These include the upper and lower limits of transformer taps, reactive power of generators, voltages at buses, switchable capacitors/reactors, and power flow over the transmission lines and transformers. The program is also supplied the values of the step changes for discrete control devices such as transformer taps, switchable capacitors and reactors. These limits are dynamically adjusted based on seasonal/weather related information. This information is in the form of whether the load is increasing or decreasing.

3. The sensitivity matrix given by equation 5.8 is formulated based on existing power system conditions.

4. The available control margins for controllers are computed as follows:
   Control margins for under-voltage = (Upper limit of controllers) - (Existing setting of controllers)
   Control margins for over-voltage = (Existing setting of controllers) - (Lower limit of controllers)

5. If the reactive power output of any generator is out of limits, that generator is taken out of the list of available controllers.

6. A check is made on the availability of each controller. A message is flashed if any controller is not available. *If all the available control margins are zero a warning is flashed on the screen. This indicates that a solution to the voltage/VAr problem cannot be found by mere manipulation of controllers.* More drastic steps such as load-shedding may be required.

7. A *violation vector* is computed by comparing existing dependent variable readings with upper and lower limits. For each dependent variable, an entry is made in the violation vector. If the reading is within the limits, an entry of *one* is made else an entry of *zero* is made. Thus a violation vector is a string of zeros and ones.

8. A check is made on the voltage deviations for all the load buses. Deviation vectors for *under-voltage* and *over-voltage* buses are made. The bus with the largest deviation (target bus) is found and a corresponding flag ('L' flag for under-voltage or 'U' for over-voltage) is flagged signaling the type of problem that needs to be solved.

9. The *best controller* is selected for the target bus. This is done by computing the product of the sensitivity of the controllers to the target bus and the available control margins. The controller that gives the maximum product of the sensitivity of the controller to the target bus and available control margin is identified. This controller is selected.

10. An appropriate increment ΔV_c in the control step of the desired controller is calculated as:

$$\Delta V_c = \Delta V_{desired}/S \tag{5.53}$$

where the desired change in voltage at the target bus is given by:

$$\Delta V_{desired} = V_{limit} - V_{desired} \tag{5.54}$$

and S is the corresponding sensitivity coefficient.

11. For discrete devices this control increment is rounded to the nearest available physical step, so that the control action is realizable.

12. The effect of the chosen controller is computed for all system parameters, and all system values are updated.

13. A new violation vector is computed and compared with the old violation vector.

14. If the new violation vector is *better* than the old violation vector, then the suggestion for this controller is accepted. A *better* new violation vector means that the number of violations is the same or smaller than before, and also, no old non-violation should become a violation after the action of the controller.

15. If the selected controller introduces new voltage violations, or if the reactive generation of the generator buses are beyond operational limits, then the control increment is reduced, until the limits are enforced. This is done by decrementing the setting of the controller by one step, and then repeating steps 12 to 14.

16. If the latest control margin for any controller is zero then this controller is removed from the list of available control actions.

17. If the target bus voltage is within the allowed limits, and the new violation vector is better than the old violation vector, then the remaining system violations are re-examined. A new target bus is declared and the program control goes to step 9. Steps 9 to 17 are repeated till all violations are removed.

18. The system exits if all voltage violations are taken care of and the reactive power of all generators are within the limits. When the available facilities are insufficient to cope with the problem, the program examines the load forecast information to see if any relief is likely within the next few minutes. If that is not the case, the program suggests adding more facilities or shedding some of the loads.

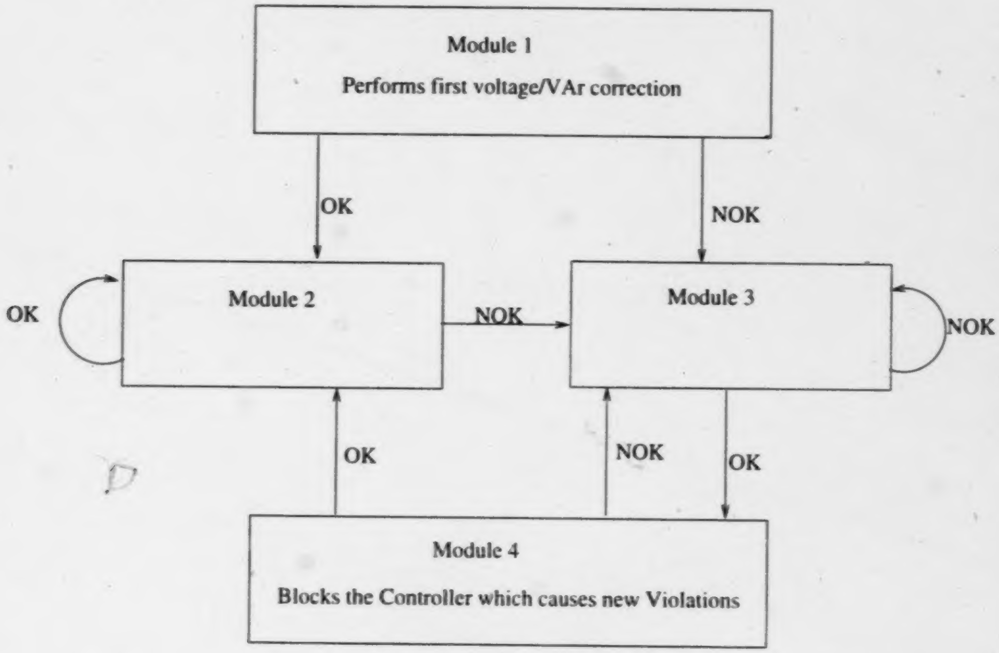

Module 1
Performs first voltage/VAr correction
OK
NOK
OK
Module 2
NOK
Module 3
NOK
OK
NOK
OK
Module 4
Blocks the Controller which causes new Violations

Figure 5.3: Modules of VVCM

19. Based on the new settings of controllers, the new power flows over the transmission lines and transformers are predicted by a neural network based system.

Note: As evident from above, the process of selection of controller settings is an iterative process. A number of book-keeping functions such as updating new controller margins, blocking controllers which introduce new violations are carried out. Figure 5.3 shows the modules that make up the VVCM. Module 1 initializes the VVCM, reads the system parameters, computes the sensitivity matrix and makes the first decision for voltage/VAr correction. Module 2 caters for cases in which a decision for voltage/VAr is accepted in one pass. Module 3 takes care of decrementing the controller value, if the suggested control action does not produce a better violation vector. Once a controller action has been decremented, it has to be blocked for further actuation (although it still may be the preferred controller based on the product of the sensitivity matrix and the available control margin). Module 4 takes care of this requirement.

The VVCM, algorithm developed as part of this research work has a number of improvements over other implementations reported in literature [Belhadj96, Cheng88, Chowdhury93, Godart91, Hsu92, Keyhani89, Liu90, Ramos95, Singh93, Yokoyama93, Zalapa95]. These are:

1. Implementation in a conventional language makes it faster.
2. Incorporation of weather/seasonal related information.
3. Ability to forecast new power flows once the setting of the controllers have been changed.
4. An efficient modular structure, as discussed above, to deal with various possibilities.

5.6 Formulation of Linear Programming for Voltage/VAr Violations

A linear programming (LP) method for voltage/VAr violation has been adapted from [Mamandur81, Hsu92]. A linear program is formulated as:

$$\text{minimize: } f(x)$$
$$\text{subject to inequality constraints: } Ax \leq b$$

where the objective function, f(x) is given by:

$$f(x) = |nt_1| + ...|nt_t| + |ng_1| + ...|ng_n| + |nq_1| + ...|nq_m| \qquad (5.55)$$

where $(|nt_1| \ ... \ |nt_t|)$, $(|ng_1|...|ng_n|)$ and $(|nq_1| \ ...|nq_m|)$ stand for absolute number of step changes for t number of transformers, n number of generators, and m number of reactors/capacitors respectively,
or

$$f(x) = nt_1 + ...nt_t + ng_1 + ...ng_n + nq_1 + ...nq_m \qquad (5.56)$$

where $(nt_1 \ ... \ nt_t)$, $(ng_1...ng_n)$ and $(nq_1 \ ...nq_m)$ stand for the number of step changes for t number of transformers, g number of generators, and m number of reactors/capacitors respectively.

The first objective function given by equation 5.55 is more restrictive than the second objective function given by equation 5.56.

$Ax <= b$ represents the inequality constraints, which are represented as follows:

$$\begin{bmatrix} \Delta Q_{g1}(min) \\ \dots \\ \Delta Q_{gn}(min) \end{bmatrix} \leq [S_{Q_gT} \quad S_{Q_gV_g} \quad S_{Q_gQ_l}] * \begin{bmatrix} \Delta T_1 \\ \dots \\ \Delta T_t \\ \Delta V_{g1} \\ \dots \\ \Delta V_{gn} \\ \Delta Q_{l1} \\ \dots \\ \Delta V_{gn} \\ \Delta Q_{l1} \\ \dots \\ \Delta Q_{lm} \end{bmatrix} \leq \begin{bmatrix} \Delta Q_{g1}(max) \\ \dots \\ \Delta Q_{gn}(max) \end{bmatrix} \tag{5.57}$$

$$\begin{bmatrix} \Delta V_{l1}(min) \\ \dots \\ \Delta V_{lm}(min) \end{bmatrix} \leq [S_{V_lT} \quad S_{V_lV_g} \quad S_{V_lQ_l}] * \begin{bmatrix} \Delta T_1 \\ \dots \\ \Delta T_t \\ \Delta V_{g1} \\ \dots \\ \Delta V_{gn} \\ \Delta Q_{l1} \\ \dots \\ \Delta Q_{lm} \end{bmatrix} \leq \begin{bmatrix} \Delta V_{l1}(max) \\ \dots \\ \Delta V_{lm}(max) \end{bmatrix} \tag{5.58}$$

$$\begin{bmatrix} \Delta T_1(min) \\ \dots \\ \Delta T_t(min) \end{bmatrix} \leq \begin{bmatrix} \Delta T_1 \\ \dots \\ \Delta T_t \end{bmatrix} \leq \begin{bmatrix} \Delta T_1(max) \\ \dots \\ \Delta T_t(max) \end{bmatrix} \tag{5.59}$$

$$\begin{bmatrix} \Delta V_{g1}(min) \\ \dots \\ \Delta V_{gn}(min) \end{bmatrix} \leq \begin{bmatrix} \Delta V_{g1} \\ \dots \\ \Delta V_{gn} \end{bmatrix} \leq \begin{bmatrix} \Delta V_{g1}(max) \\ \dots \\ \Delta V_{gn}(max) \end{bmatrix} \tag{5.60}$$

$$\begin{bmatrix} \Delta Q_{l1}(min) \\ \dots \\ \Delta Q_{lm}(min) \end{bmatrix} \leq \begin{bmatrix} \Delta Q_{l1} \\ \dots \\ \Delta Q_{lm} \end{bmatrix} \leq \begin{bmatrix} \Delta Q_{l1}(max) \\ \dots \\ \Delta Q_{lm}(max) \end{bmatrix} \tag{5.61}$$

where

$$\begin{bmatrix} \Delta Q_{g1}(min) \\ \ldots \\ \Delta Q_{gn}(min) \end{bmatrix} = \begin{bmatrix} Q_{g1} - Q_{g1}(min) \\ \ldots \\ Q_{gn} - Q_{gn}(min) \end{bmatrix} \tag{5.62}$$

$$\begin{bmatrix} \Delta Q_{g1}(max) \\ \ldots \\ \Delta Q_{gn}(max) \end{bmatrix} = \begin{bmatrix} Q_{g1}(max) - Q_{g1} \\ \ldots \\ Q_{gn}(max) - Q_{gn} \end{bmatrix} \tag{5.63}$$

$$\begin{bmatrix} \Delta V_{l1}(min) \\ \ldots \\ \Delta V_{lm}(min) \end{bmatrix} = \begin{bmatrix} V_{l1} - V_{l1}(min) \\ \ldots \\ V_{lm} - V_{lm}(min) \end{bmatrix} \tag{5.64}$$

$$\begin{bmatrix} \Delta V_{l1}(max) \\ \ldots \\ \Delta V_{lm}(max) \end{bmatrix} = \begin{bmatrix} V_{l1}(max) - V_{l1} \\ \ldots \\ V_{lm}(max) - V_{lm} \end{bmatrix} \tag{5.65}$$

$$\begin{bmatrix} \Delta T_1(min) \\ \ldots \\ \Delta T_t(min) \end{bmatrix} = \begin{bmatrix} T_1 - T_1(min) \\ \ldots \\ T_t - T_t(min) \end{bmatrix} \tag{5.66}$$

$$\begin{bmatrix} \Delta T_1(max) \\ \ldots \\ \Delta T_t(max) \end{bmatrix} = \begin{bmatrix} T_1(max) - T_1 \\ \ldots \\ T_t(max) - T_t \end{bmatrix} \tag{5.67}$$

$$\begin{bmatrix} \Delta V_{g1}(min) \\ \ldots \\ \Delta V_{gn}(min) \end{bmatrix} = \begin{bmatrix} V_{g1} - V_{g1}(min) \\ \ldots \\ V_{gn} - V_{gn}(min) \end{bmatrix} \tag{5.68}$$

$$\begin{bmatrix} \Delta V_{g1}(max) \\ \ldots \\ \Delta V_{gn}(max) \end{bmatrix} = \begin{bmatrix} V_{g1}(max) - V_{g1} \\ \ldots \\ V_{gn}(max) - V_{gn} \end{bmatrix} \tag{5.69}$$

$$\begin{bmatrix} \Delta Q_{l1}(min) \\ \ldots \\ \Delta Q_{lm}(min) \end{bmatrix} = \begin{bmatrix} Q_{l1} - Q_{l1}(min) \\ \ldots \\ Q_{lm} - Q_{lm}(min) \end{bmatrix} \tag{5.70}$$

$$\begin{bmatrix} \Delta Q_{l1}(max) \\ \ldots \\ \Delta Q_{lm}(max) \end{bmatrix} = \begin{bmatrix} Q_{l1}(max) - Q_{l1} \\ \ldots \\ Q_{lm}(max) - Q_{lm} \end{bmatrix} \tag{5.71}$$

5.6.1 Mixed Linear Integer Programming

Integer linear program is a linear programming (LP) problem, wherein some or all the decision variables are restricted to be integer values. A *pure integer program* is one in which all the variables are restricted to be integers. A *mixed integer program* restricts some of the variables to be integers while others can assume continuous (fractional) values. Mixed integer linear program is formulated as:

$$\text{minimize f(x)}$$
$$\text{subject to : } Ax \leq b$$

where f(x) is the objective function, given by equation 5.55 or 5.56.
$Ax \leq b$ represents the constraints, given by equations 5.63 to 5.71.
An additional constraint is that:
x_j's are integer and x_j are elements of $\{(nt_1 \ldots nt_t), (ng_1 \ldots ng_n), (nq_1 \ldots nq_m)\}$

5.7 Formulation of Load-Shedding Methods

Three load-shedding methods have been formulated.

5.7.1 Load-Shedding Method 1 (LSM-SC)

This load-shedding method has been developed using the sensitivity coefficients. This method is abbreviated as (LSM-SC) which stands for load-shedding method based on sensitivity coefficients. The use of sensitivity coefficients for load-shedding has also been suggested by [Keyhani89, Singh93]. However, the way the sensitivity coefficients have been used in this research (for load-shedding) has not been reported in the literature. The following voltage vectors are defined:

$V_l(min)$ = minimum voltage desired on load buses

$V_l(nominal)$ = nominal voltage desired on load buses (generally 1 p.u.)

$V_l(normal)$ = normal (with the specified upper and lower limits) voltage desired on load buses

$V_l(max)$ = maximum voltage desired on load buses

V_l = present load bus voltages

For load-shedding calculations, the calculations proceed by computing four voltage deviation vectors for the load buses. They are:

1. $\Delta V_l(min)$ which is given by:

$$\Delta V_l(min) = V_l(min) - V_l \tag{5.72}$$

2. $\Delta V_l(nominal)$ which is given by:

$$\Delta V_l(nominal) = V_l(nominal) - V_l \tag{5.73}$$

3. $\Delta V_l(normal)$ which is given by:

for $(V_l < V_l(min))$

$$\Delta V_l(normal) = V_l(nominal) - V_l \tag{5.74}$$

else

$$\Delta V_l(normal) = 0 \tag{5.75}$$

4. $\Delta V_l(max)$ which is given by:

$$\Delta V_l(max) = V_l(max) - V_l \tag{5.76}$$

From equation 5.8 it is noted that if ΔT and ΔV_g's are set equal to zero, one gets:

$$\begin{bmatrix} \Delta Q_{g1} \\ \cdots \\ \Delta Q_{gn} \\ \Delta V_{l1} \\ \cdots \\ \Delta V_{lm} \end{bmatrix} = \begin{bmatrix} S_{QgQl} \\ S_{VlQl} \end{bmatrix} \begin{bmatrix} \Delta Q_{l1} \\ \cdots \\ \Delta Q_{lm} \end{bmatrix} \tag{5.77}$$

Expanding the above equation, one gets:

$$\begin{bmatrix} \Delta Q_{g1} \\ \cdots \\ \Delta Q_{gn} \end{bmatrix} = [S_{QgQl}] \begin{bmatrix} \Delta Q_{l1} \\ \cdots \\ \Delta Q_{lm} \end{bmatrix} \tag{5.78}$$

and

$$\begin{bmatrix} \Delta V_{l1} \\ \cdots \\ \Delta V_{lm} \end{bmatrix} = [S_{VlQl}] \begin{bmatrix} \Delta Q_{l1} \\ \cdots \\ \Delta Q_{lm} \end{bmatrix} \tag{5.79}$$

The above equation is abbreviated as:

$$[\Delta V_l] = [S_{VlQl}][\Delta Q_l] \tag{5.80}$$

From the above equation one can compute $[\Delta Q_l]$, the amount of load which must be shed to recover the voltages:

$$[\Delta Q_l] = [S_{VlQl}]^{-1}[\Delta V_l] \tag{5.81}$$

For the four delta voltages, given by equations 5.72 to 5.76 compute corresponding load-sheds, i.e. $\Delta\ Q_l(min)$ for $\Delta\ V_l(min)$; $\Delta\ Q_l(nominal)$ for $\Delta\ V_l(nominal)$; $\Delta\ Q_l(normal)$ for $\Delta\ V_l(normal)$; and $\Delta\ Q_l(max)$ for $\Delta\ V_l(max)$.

The $\Delta Q_l(normal)$ and $\Delta Q_l(max)$ values provide the safe range within which the loads may be shed. The four values of load-sheds recommended present alternative scenarios for the operator. Corresponding to reactive power load-sheds, real power loads may also be shed.

5.7.2 Load-Shedding Method 2 (LSM-LP)

This linear programming formulation for load-shedding has been adapted from the linear programming method for voltage and VAr management suggested by [Mamandur82]. This method is abbreviated as (LSM-LP) which stands for load-shedding method based on linear programming. A linear programming module for load-shedding is formulated as follows:

$$\text{minimize } f(x)$$

$$\text{subject to the inequality constraints: } Ax \leq b$$

where the objective function is given by:

$$f(x) = |nq_1| + ...|nq_m| \tag{5.82}$$

where $(|nq_1| \; ...|nq_m|)$ stand for the absolute number of step changes for m number of reactive loads.

or

$$f(x) = nq_1 + ...nq_m \tag{5.83}$$

where $(nq_1 \; ...nq_m)$ stand for the number of step changes for m number of reactive loads.

The first objective function given by equation 5.82 is more restrictive than the second objective function given by equation 5.83.

$Ax <= b$ represents the constraints, which are represented as follows:

$$\begin{bmatrix} \Delta Q_{g1}(min) \\ ... \\ \Delta Q_{gn}(min) \end{bmatrix} \leq [S_{Q_gQ_l}] * \begin{bmatrix} \Delta Q_{l1} \\ ... \\ \Delta Q_{lm} \end{bmatrix} \leq \begin{bmatrix} \Delta Q_{g1}(max) \\ ... \\ \Delta Q_{gn}(max) \end{bmatrix} \tag{5.84}$$

$$\begin{bmatrix} \Delta V_{l1}(min) \\ ... \\ \Delta V_{lm}(min) \end{bmatrix} \leq [S_{V_lQ_l}] * \begin{bmatrix} \Delta Q_{l1} \\ ... \\ \Delta Q_{lm} \end{bmatrix} \leq \begin{bmatrix} \Delta V_{l1}(max) \\ ... \\ \Delta V_{lm}(max) \end{bmatrix} \tag{5.85}$$

$$\begin{bmatrix} 0 \\ \dots \\ 0 \end{bmatrix} \leq \begin{bmatrix} \Delta Q_{l1} \\ \dots \\ \Delta Q_{lm} \end{bmatrix} \leq \begin{bmatrix} Q_{l1} \\ \dots \\ Q_{lm} \end{bmatrix} \tag{5.86}$$

where

$$\begin{bmatrix} \Delta Q_{g1}(min) \\ \dots \\ \Delta Q_{gn}(min) \end{bmatrix} = \begin{bmatrix} Q_{g1} - Q_{g1}(min) \\ \dots \\ Q_{gn} - Q_{gn}(min) \end{bmatrix} \tag{5.87}$$

$$\begin{bmatrix} \Delta Q_{g1}(max) \\ \dots \\ \Delta Q_{gn}(max) \end{bmatrix} = \begin{bmatrix} Q_{g1}(max) - Q_{g1} \\ \dots \\ Q_{gn}(max) - Q_{gn} \end{bmatrix} \tag{5.88}$$

$$\begin{bmatrix} \Delta V_{l1}(min) \\ \dots \\ \Delta V_{lm}(min) \end{bmatrix} = \begin{bmatrix} V_{l1} - V_{l1}(min) \\ \dots \\ V_{lm} - V_{lm}(min) \end{bmatrix} \tag{5.89}$$

$$\begin{bmatrix} \Delta V_{l1}(max) \\ \dots \\ \Delta V_{lm}(max) \end{bmatrix} = \begin{bmatrix} V_{l1}(max) - V_{l1} \\ \dots \\ V_{lm}(max) - V_{lm} \end{bmatrix} \tag{5.90}$$

The last constraint given by equation 5.86 specifies that the load-shed has to be greater than or equal to zero and less than or equal to existing reactive loads.

5.7.3 Load-Shedding Method 3 (LSM-DI)

This method has been developed as part of this research. This method is abbreviated as (LSM-DI) which stands for load-shedding method based on **D**elta current (**I**) injection method. The method may be explained by considering a two bus system as shown in Figure 5.3. Subscripts 1 and 2 refer to the generator and load bus respectively.

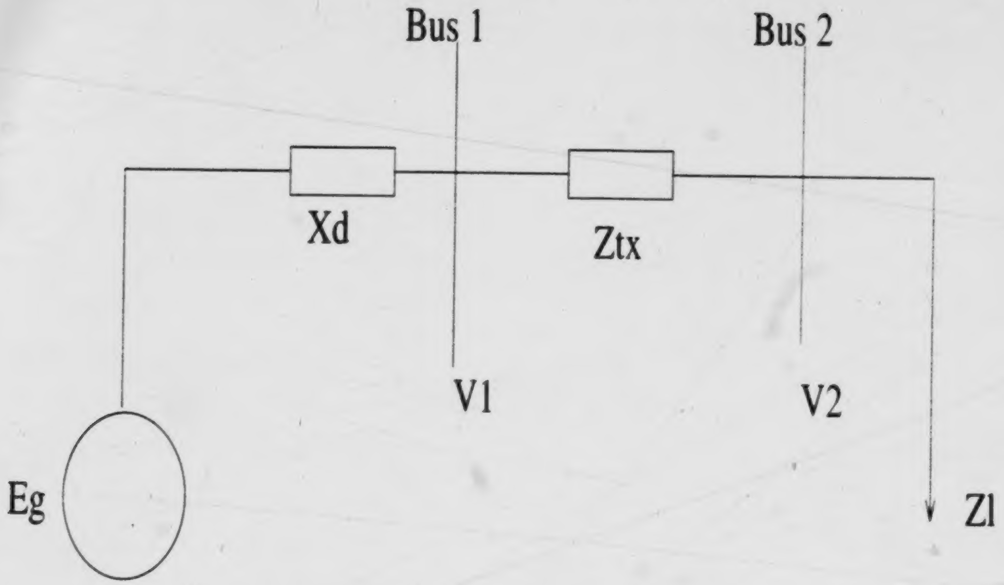


Figure 5.4: Two Bus Power System

The steps in the algorithm for load-shedding are:

1. Compute the admittance matrix $\hat{Y}_{BUS}$ of the system.

2. Compute the desired change in voltage. $\Delta\hat{V}$ = (desired voltage - existing voltage) for all the buses. $\hat{V}$ is the vector giving the existing values of voltages.

3. Compute the desired change in current. $\Delta\hat{I}$ for all the buses using the formula:

$$\Delta\hat{I} = \hat{Y}_{BUS} * \Delta\hat{V} \tag{5.91}$$

   For the load bus, the delta current injections may be replaced by an equivalent impedance:

$$\Delta\hat{Z} = -(\hat{V} + \Delta\hat{V})/(\Delta\hat{I}) \tag{5.92}$$

4. For the generator bus two possibilities exist:

   (a) **Regulating the Generator Voltage**

      - Calculate the old generator current. $\hat{I}_{go}$:

$$\hat{I}_{go} = \hat{E}_{go}/\hat{X}_d \tag{5.93}$$

where

$\hat{X}_d$ is the reactance of the generator, and

$\hat{E}_{go}$ is the old generator voltage

- Calculate the new generator current, $\hat{I}_{gn}$:

$$\hat{I}_{gn} = \hat{I}_{go} + \Delta \hat{I}_1 \tag{5.94}$$

- Calculate the new generator voltage, $\hat{E}_{gn}$:

$$\hat{E}_{gn} = \hat{I}_{gn} * \hat{X}_d \tag{5.95}$$

This method allows one to find the effect of impedance change at the load buses on the generator voltage. For proper operation the generator voltage should be within the specified limits, after the load-shedding has been implemented.

(b) **Introducing Impedance Compensation at the Generator Bus**

In this method, a compensation impedance $\Delta\hat{Z}_1$ is introduced in parallel with $(\hat{Z}_{ln} + \hat{Z}_{tr})$

where

$\hat{Z}_{ln}$ is the new load, and

$\hat{Z}_{tr}$ is the transmission line impedance.

The new load $\hat{Z}_{ln}$ is given by:

$$\hat{Z}_{ln} = (\Delta\hat{Z}_2 * \hat{Z}_{lo})/(\Delta\hat{Z}_2 + \hat{Z}_{lo}) \tag{5.96}$$

where

$\Delta\hat{Z}_2$ is the compensation impedance at bus 2

$= (\hat{V}_2 + \Delta\hat{V}_2)/\Delta\hat{I}_2$

and

$\hat{Z}_{lo}$ is the old load impedance.

This method can only be applied when there is a possibility of changing the impedance at the generator bus.

5.7.4 Comments on Load-Shedding Methods

Three methods for load-shedding have been formulated. Method 1, (LSM-SC) is computationally least intensive as it is based upon the inverse of appropriate Sensitivity Coefficients, which can be computed before voltage stability problems begin to be serious. The use of sensitivity coefficients for load-shedding has been contemplated by [Keyhani89, Singh93] but not in the way it has been presented in this thesis. [Liu90, Yokoyama93] have argued against the use of sensitivity coefficients for load-shedding, on account of inaccuracies. In this author's opinion, if sensitivities are calculated based on conditions just before load-shedding, the results are satisfactory, as shown in the next chapter. Method 2, (LSM-LP) can be computationally intensive depending on the number of variables as it is based on Linear Programming methodology. However, it may be pointed out that the number of variables for load-shedding are much less as compared to the voltage/VAr violation problem. Method 3 (LSM-DI) based on Δ I (current) injections allows to take into account the effect of both real and reactive power for load-shedding calculations, but needs further research.

5.8 Generation of Database for Neural Network and Decision Tree

In general, obtaining a representative database for training of neural networks and decision trees is a difficult problem. Whereas for many learning problems, one may consider that the data bases are provided a priori (for example, load forecasting, letter recognition, etc.), for the problem at hand the samples need to be generated via a computer-based simulation of random sampling. This is because it is neither possible nor desirable to build these samples solely by collecting data from normal power system situations [Wehenkel93].

Algorithm for Generating the Database

As noted above, it is necessary to generate a database for power system operation by simulation. The following algorithm, developed as a part of this thesis, allows a wide range of power system operating scenarios to be incorporated.

1. **Step 1:** Run the base case load flow.

2. **Step 2:** Run the VQSM program which determines if the system is voltage stable.

3. **Step 3:** If the case is voltage stable, record values of the control and dependent variables, the *voltage - reactive power - sensitivity* vector and that the case is stable. If the system is unstable, record the values of the control variables and that the case is unstable.

4. **Step 4:** Check the case number. If the case number is less then or equal to the maximum number of cases, then go to step 5; else go to step 7.

5. **Step 5:** Change the system parameters, that is the settings of transformer taps, voltage setting of generators, generated and load real and reactive powers.

6. **Step 6:** Run the load flow. Go to step 2.

7. **Step 7:** Exit.

5.8.1 Generation of New Values of Variables

This section shows how new values are generated for tap ratios of transformers, generator voltages, real and reactive loads, and generations.

Generation of New Values of Generator Voltage

Step 1: Generate a random number = x, (x = 0 to 0.9999);
Step 2: Multiply this random number with old generator voltage, $V_g(old)$
$V_g(new) = V_g(old)*x$;
Step 3: Check that the new generator voltage, $V_g(new)$ is within the limits:

```
If ($V_g(new) < V_g(lower)$)
        $V_g(new) = V_g(lower) + x/factor$;
elseif ($V_g(new) > V_g(upper)$)
        $V_g(new) = V_g(upper) - x/factor$;
else $V_g(new) = V_g(old)*x$;
end
```

Note: A typical $V_g(lower)$ value is 0.9;
A typical $V_g(upper)$ value is 1.1;
A typical value of "factor" is 5.

Generation of New Values of Transformer Tap Ratios

Step 1: Generate a random number = x, (x = 0 to 0.9999);
Step 2: Multiply this random number with old tap ratio, tap_ratio(old);

```
    tap_ratio(new) = tap_ratio(old)*x;
```

Step 3: Check that the new tap ratio is within the limits:

```
    If (tap_ratio(new) < tap(lower))
            tap_ratio(new) = tap(lower) + x/factor;
    elseif (tap_ratio(new) > tap(upper))
            tap_ratio(new) = tap(upper) - x/factor;
    else tap_ratio(new) = tap_ratio(old)*x;
    end
```

Step 4: The tap_ratio(new) value is rounded off to reflect the discrete nature of the transformer tap ratios.

Note: A typical tap_ratio(lower) value is 0.8;
A typical tap_ratio(upper) value is 1.20;
A typical value of "factor" is 5;
Typically, tap ratio values are rounded off to increments of 0.01.

Generation of New Values of Real Power (Load)

Step 1: Generate a random number = x, (x = 0 to 0.9999);
Step 2: Generate new value of real power load, $P_l(new)$:

```
    If ($P_l(old)$ < 0.001)
            $P_l(new)$ = x*load_factor; (OPTIONAL)
    else $P_l(new)$ = $P_l(old)$*x*load_factor;
    end
```

Note: A typical value of "load_factor" is a measure of how much the system can be overloaded over the normal load;
OPTIONAL indicates that there is a provision to introduce real power loads at buses which had no real power load previously.

Generation of New Values of Reactive Power (Load)

Step 1: Generate a random number = x, (x = 0 to 0.9999);

Step 2: Generate new value of reactive power load, $Q_l(new)$:

```
If ($Q_l(old) < 0.001$)
        $Q_l(new) =$ x*load_factor; (OPTIONAL)
else $Q_l(new) = Q_l(old)$*x*load_factor;
end
```

Note: A typical value of "load_factor" is a measure of how much the system can be overloaded over the normal load;

OPTIONAL indicates that there is a provision to introduce reactive power loads at buses which had no reactive power load previously.

Generation of New Values of Real Power (Generation)

Step 1: Generate a random number = x, (x = 0 to 0.9999);

Step 2: Generate new value of real power generation, $P_g(new)$:

```
If ($P_g(old) < 0.001$)
        $P_g(new) =$ x*load_factor; (OPTIONAL)
else $P_g(new) = P_g(old)$*x*load_factor;
end
```

Step 3: For the slack bus, P_g(new) =0

Note: A typical value of "load_factor" is a measure of how much the system can be overloaded over the normal load;

OPTIONAL indicates that there is a provision to introduce real power generations at buses which had no real power generations previously.

Generation of New Values of Reactive Power (Generation)

Step 1: Generate a random number = x, (x = 0 to 0.9999);

Step 2: Generate new value of reactive power generation, $Q_g(new)$:

```
If $(Q_g(old) < 0.001)$
        $Q_g(new)$ = x*load_factor; (OPTIONAL)
else $Q_g(new) = Q_g(old)$*x*load_factor;
end
```

Step 3: For all generator buses, Q_g(new) =0

Note: A typical value of "load_factor" is a measure of how much the system can be overloaded over the normal load;
OPTIONAL indicates that there is a provision to introduce reactive power generations at buses which had no reactive power generation previously.

5.9 Training of Neural Network

In this thesis, a neural network has been trained for predicting new power flows over the transmission lines and transformers, once the setting of controllers has changed. For this purpose, 1000 to 2000 data sets are simulated by using algorithms in the previous sub-section. The output data (power flows over transmission lines and transformers) is pre-processed by classifying under five levels.

The input for predicting power flows (for example, between buses i and j) comprises of the following:

$$input = [(tap_1...tap_t); (V_i - V_j); (P_{gi} - P_{li}); (Q_{gi} - Q_{li}); (P_{gj} - P_{lj}); (Q_{gj} - Q_{lj})]; \quad (5.97)$$

where
tap_1 ... tap_t are the tap settings of t number of transformers
(V_i - V_j) is the voltage difference between buses i and j
(P_{gi} - P_{li}) is the difference in real power generation and real power load at bus i
(Q_{gi} - Q_{li}) is the difference in reactive power generation and reactive power load at bus i
(P_{gj} - P_{lj}) is the difference in real power generation and real power load at bus j
(Q_{gj} - Q_{lj}) is the difference in reactive power generation and reactive power load at bus j

The output is level of power flows over the transmission line or transformer connecting buses i and j.

The neural network architecture used in this thesis is ADALINE (**ADA**ptive **LI**near **NE**uron [Patterson96]. For the input and output defined above, this neural-network architecture uses one layer of neuron connected to (5 + no. of transformers) inputs through a matrix of weights, W. The number of neurons is equal to (5 + no. of transformers). This neural network architecture uses Widrow-Hoff Learning Algorithm [Widrow85]. This algorithm, which is a form of supervised learning, is also known as the least mean square (LMS) method or the Delta Rule. For a given training input pattern x, the learning rule is given by the iterative formula:

$$w_i^{new} = w_i^{old} + \alpha(t - y)x_i \tag{5.98}$$

where α is a learning coefficient, t is the desired or target output value for input vector x, and y is the actual value computed by the neuron.

5.10 Inducing a Decision Tree

The training set consists of tuples that comprises transformer tap settings, generator voltages and all real and reactive loads. All the input values are continuous. The induced decision tree classifies the power system into *Stable*, *IntStable* (intermediate stable) or *Unstable* mode depending upon the input pattern. The class *IntStable* was classified by examining the load bus voltages and the corresponding **V**oltage reactive power **Q** **S**ensitivity (VQS) values. For VQS values above a certain limit (with load flow converged) the power system operating mode was classified as *IntStable*.

Once a decision tree has been induced, the VQS method (as shown in Figure 5.1) is replaced by the DM module as shown in Figure 5.5.

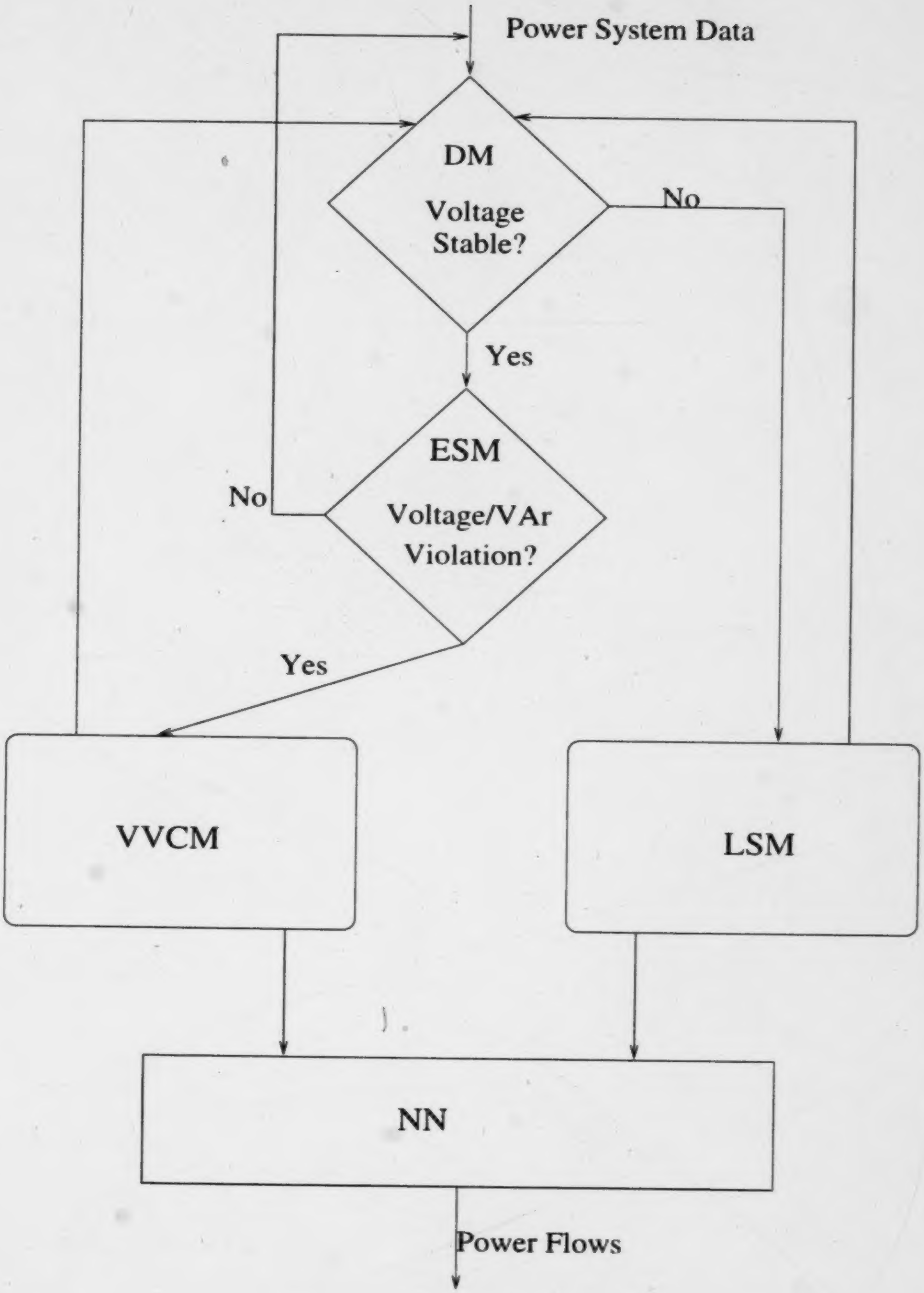

Power System Data
DM
Voltage Stable?
No
Yes
ESM
Voltage/VAr Violation?
No
Yes
VVCM
LSM
NN
Power Flows

Figure 5.5: Structure of CVVC Toolbox

5.11 Conclusion to Chapter 5

This chapter has presented the models and algorithms that have been used to obtain results for this thesis. In this work, sensitivities were calculated by using complex impedance rather than only the imaginary impedance of the system. For tap changing transformers, a new model has been proposed, in this thesis, which gives better results than those that can be obtained using already existing models. The VVCM module has been implemented primarily in MATLAB because the VVCM algorithm reveals that most of the functions are sequential. As such, this kind of implementation in AI languages (e.g. Prolog) can be slow. Three methods (with new features developed in this thesis) for load-shedding have been presented. A program to generate a database for the training of neural networks and inducing decision trees has also been outlined.

Chapter 6

RESULTS

This chapter describes applications of the techniques developed in this thesis to three test systems. Voltage/VAr violations and voltage stability cases are investigated for a 6 bus system [Wallach86], IEEE 14 bus system [Glover94] and IEEE 30 bus [Glover94] system. The data for these power systems is given in Appendix 'C'.

6.1 Voltage VAr Controller Method(VVCM)

Background

In this section the performance of the VVCM algorithm is evaluated for voltage/VAr violations. In this section load-shedding is not contemplated.

The following equation was derived in the previous chapter showing the relationship between changes in the controller elements and their corresponding effect on the dependent variables. The controller variables are transformers with taps, generator voltages, and switchable shunt capacitors/reactors. The dependent variables are the reactive power of the generators, and the voltages at the load buses. Considering t number of transformers with taps, n number of generators, and m number of load buses, one can write:

$$\begin{bmatrix} \Delta Q_{g1} \\ \ldots \\ \ldots \\ \Delta Q_{gn} \\ \Delta V_{l1} \\ \ldots \\ \Delta V_{lm} \end{bmatrix} = \begin{bmatrix} S_{QgT} & S_{QgVg} & S_{QgQl} \\ S_{VlT} & S_{VlVg} & S_{VlQl} \end{bmatrix} \begin{bmatrix} \Delta T_1 \\ \ldots \\ \Delta T_t \\ \Delta V_{g1} \\ \ldots \\ \Delta V_{gn} \\ \Delta Q_{l1} \\ \ldots \\ \Delta Q_{lm} \end{bmatrix} \tag{6.1}$$

where

subscripts g and l stand for the generator (PV) and load (PQ) buses respectively. The above equation may be abbreviated as:

$$[\Delta A] = [S][\Delta X] \tag{6.2}$$

where

$[\Delta X]$ is the difference vector of setting of the controllers, that is, [(X(new) - X(old)];

$[\Delta A]$ is the difference vector of value of the dependent variables, that is, [(Anew) - (Aold)], and

[S] is the sensitivity matrix which is a collection of 6 sub-matrices and links the increments of generator reactive power outputs and the load bus voltages caused by the incremental change in controllers.

The VVCM, linear programming method 1 and 2 (LP1 and LP2) have been developed in this thesis. These programs are concerned with the selection of the setting of the controllers to remedy voltage/VAr violations.

Abbreviations

Referring to equation 6.2, in the following description various "X" refer to the control variables and various "A" refer to the dependent variables. Specifically:

- X(min) = Minimum value of the control variables, that is, the minimum magnitude of the control variables
- X(old) = Value of the control variables before any corrective action is taken
- X(VVCM) = Value of the control variables suggested by the Voltage VAr Controller Method (VVCM)
- X(LP1) = Value of the control variables suggested by the linear programming method with the objective function to be minimized given by:

$$f(x) = nt_1 + ...nt_t + ng_1 + ...ng_n + nq_1 + ...nq_m \tag{6.3}$$

where $(nt_1 \; ... \; nt_t)$, $(ng_1 ... ng_n)$ and $(nq_1 \; ... nq_m)$ stand for number of step changes for t number of transformers, n number of generators, and m number of reactors/capacitors respectively.

- X(LP2) = Value of the control variables suggested by the linear programming method with the objective function to be minimized given by:

$$f(x) = |nt_1| + ...|nt_t| + |ng_1| + ...|ng_n| + |nq_1| + ...|nq_m| \quad (6.4)$$

  where $|nt_1|$ ... $|nt_t|$, $|ng_1|...|ng_n|$ and $|nq_1|$ ...$|nq_m|$ stand for absolute number of step changes for t number of transformers, n number of generators, and m number of reactors/capacitors respectively.

- X(Reference) = Value of the control variables as given in the literature for the same system

- X(max) = Maximum value of the control variables, that is, the maximum magnitude of the control variables

- A(min) = Minimum value of the dependent variables

- A(old) = Value of the dependent variables before any corrective action is taken

- A(VVCM)= Value of the dependent variables predicted by the Voltage VAr Controller Method (VVCM) after making the recommended action

- A(LP1) = Value of the dependent variables predicted after taking actions as suggested by the linear programming method with the objective function given by equation 6.3.

- A(LP2) = Value of the dependent variables predicted after taking actions as suggested by the linear programming method with the objective function given by equation 6.4.

- A(Reference) = Value of the dependent variables as given in the literature for the same system

- A(max) = Maximum value of the dependent variables

In the following tables $\Sigma\Delta X$ refer to sum of changes in setting of the control variables required to remove voltage/VAr violations, that is:

$$\Sigma\Delta X = \sum_{i=1}^{t+n+m} (X_i(new) - X_i(old)) \quad (6.5)$$

for t number of transformers, n number of generators, and m number of reactors/capacitors.

Since some control devices are discrete, their control actions are rounded to the nearest step, so that the control action is realizable. **NSTEPS(R)** (standing for number of steps which have been rounded up) gives the sum of changes in the setting of the control variables required to remove voltage/VAr violations after the control actions of discrete devices have been rounded up.

$$NSTEPS(R) = \sum_{i=1}^{t}(X_i(new) - X_i(old))/ts + \sum_{i=1}^{n}(X_i(new) - X_i(old))/v_g s + \sum_{i=1}^{m}(X_i(new) - X_i(old))/q_l s \tag{6.6}$$

where

ts is the step length for transformers tap ratio

$v_g s$ is the step length for generator voltage

$q_l s$ is the step length for switchable reactor/capacitor.

1. **Case VVC1:** This case is for a 6 bus system. A comparison is made with results from [Mamandur82], who solves this problem by linear programming. The system is at full-load level and has an under-voltage violation at bus number 3 as seen from A(old) values in Table 6.2. To remedy this situation Table 6.1 presents the recommended values of the setting of the control variables for different techniques. The techniques are:
   - VVCM and settings of the controllers is given under X(VVCM).
   - Linear Programming with objective function given by equation 6.3 and settings of the controllers is given under X(LP1).
   - Linear Programming with objective function given by equation 6.4 and setting of the controllers is given under X(LP2).
   - [Mamandur82]'s solution and setting of the controllers is given under X([Mamandur82]).

In this thesis, the solution for voltage/VAr violations using linear programming with objective function given by equation 6.4, that is, method LP2 resembles [Mamandur82]'s formulation. However, since the sensitivities calculated in this research work are different from [Mamandur82] the final solutions are sometimes different.

NSTEPS(R) is computed from $\Sigma\Delta X$ assuming controller step length of 0.01 p.u. for all types of controllers.

S. No.	Type	X(min)	X(old)	X(VVCM)	X(LP1)	X(LP2)	X([Mamandur82])	X(max)
1.	Tran 1: t	0.900	1.0250	1.0250	$\underline{0.9000}$	1.0250	1.025	1.100
2.	Tran 2: t	0.900	1.1000	1.1000	$\underline{0.9000}$	$\underline{1.0804}$	$\underline{1.092}$	1.100
3.	Gen 1: V	1.000	1.0500	$\underline{1.1000}$	$\underline{1.0958}$	$\underline{1.1000}$	$\underline{1.100}$	1.100
4.	Gen 2: V	1.000	1.1000	1.1000	$\underline{1.0000}$	1.1000	1.100	1.150
5.	Q:1(3)	0.000	0.0000	$\underline{0.0300}$	0.0000	0.0000	0.00	0.050
6.	Q:2(6)	0.000	0.0000	0.0000	0.0000	0.0000	0.00	0.055
	$\Sigma\Delta X$:			0.08	0.4708	0.0696	0.058	
	NSTEPS(R):			8	48	7	6	

Table 6.1: Control Variables (Case VVC1)

The entries in Table 6.1 are explained below:

- *Serial number (S. No.)* numerically identifies the controllers;
- *Type* identifies the type of the controller. Thus, controllers 1, 2 are tap changing transformers; 3, 4 are generators; and 5, 6 are capacitors. The number in bracket of controllers at serial numbers 5 and 6 indicate the bus numbers at which they are installed. Thus Q:1(3) means that the first capacitor is installed at bus number 3 and Q:2(6) means that the second capacitor is installed at bus number 6;
- X(min) is the minimum value of the control variables;
- X(old) is the value of the control variables before any corrective action is taken;
- X(VVCM), X(LP1), X(LP2) are values of the control variables recommended by the VVCM, LP1 (linear programming method 1) and LP2 (linear programming method 2) respectively;
- X([Mamandur82]) gives value of the control variables based on study in [Mamandur82];
- X(max) is the maximum value of the control variables.

In Table 6.1, the *underlined* values in columns under X(VVCM), X(LP1), X(LP2) and X([Mamandur82]) show values of the control variables which have changed from X(old). For examples, on examining, columns under X(VVCM) one notes that entries at serial numbers 3 and 5 are *underlined* as the VVCM has suggested that the generator voltage (at bus 2) be increased from 1.05 to 1.1 p.u. and a capacitance of value 0.03 p.u. be switched on (at bus 3).

$\Sigma\Delta X$ under VVCM is calculated as follows:

$$\Sigma\Delta X(VVCM) = \sum_{i=1}^{t+n+m} (X_i(VVCM) - X_i(old))$$

From entries in X(old) and X(VVCM) changes in controller settings are 0.05 p.u. and 0.03 p.u. at serial numbers 3 and 5 respectively. Hence $\Sigma\Delta X(VVCM)$ is 0.08 p.u..

NSTEPS(R) under the VVCM is calculated as follows:

As stated earlier for this example, a controller step length of 0.01 p.u. was assumed for all types of controllers. Hence, 5 steps are needed to move the generator voltage at bus 2 from 1.05 to 1.10 p.u., and 3 step are needed to switch 0.03 p.u. capacitance at bus 3. Therefore, NSTEP(R) for VVCM is 8. The above calculations are repeated for X(LP1), X(LP2) and X([Mamandur82]). Values of NSTEP(R) for LP1, LP2 and [Mamandur82] are 48, 7 and 6 respectively.

One notes, that the VVCM provides a comparable solution with [Mamandur82] and LP2 methods, based on the criteria of minimizing the number of changes in the controller settings, that is NSTEPS(R). The time required[1] to solve this problem by the VVCM method is less as compared to methods involving linear programming especially with the objective function given by equation 6.4 which is also the method used by [Mamandur82].

Figure 6.1 presents a subset of information in Table 6.1 graphically. This figure enables an examination of how the controllers are adjusted by various techniques for remedying voltage/VAr violations.

Note: In all the bar graphs the top entry in the legend refers to the left most bar and the bottom entry in the legend refers to the right most bar.

---

[1] This statement is based on the throughput time required to solve the problem by different methods. A more accurate quantitative measure would be the CPU time required after the optimization of codes under comparison.

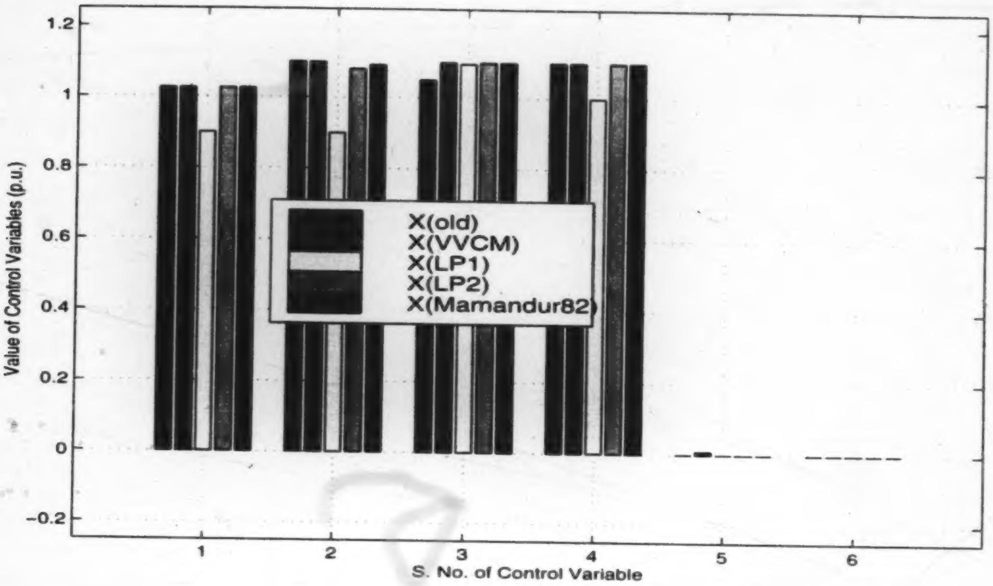


Figure 6.1: Control Variables (Case VVC1)

Table 6.2 gives the predicted values of the dependent variables for the techniques described above. These predictions are made by using the sensitivities of the dependent variables w.r.t. the control variables as derived in the previous chapter.

Type	A(min)	A(old)	A(VVCM)	A(LP1)	A(LP2)	A([Mamandur82])	A(max)
Bus 1: Q	-0.1955	0.3812	0.4082	0.7209	0.4407	0.4207	1.0004
Bus 2: Q	-0.1955	0.3481	0.2683	-0.0223	0.2697	0.2564	1.0004
Bus 3: V	0.9004	0.8552	0.8989	0.9598	0.9002	0.900	1.0004
Bus 4: V	0.9004	0.9525	0.9997	0.9317	0.9909	0.999	1.0004
Bus 5: V	0.9004	0.9001	0.9289	0.9248	0.9261	0.931	1.0004
Bus 6: V	0.9004	0.9332	0.9734	0.9004	0.9693	0.978	1.0004

Table 6.2: Dependent Variables (Case VVC1)

The entries in Table 6.2 are explained below:

- *Type* identifies the type of the dependent variables. Thus, dependent variables 1, 2 are the reactive power of generators; and 3 to 6 are voltages at the load buses;
- A(min) is the minimum value of the dependent variables;
- A(old) is the value of the dependent variables before any corrective action is taken;

- A(VVCM), A(LP1), A(LP2) are the predicted values of the dependent variables after taking actions recommended by the VVCM, linear programming method 1 (LP1), and linear programming method 2 (LP2) respectively;
- A([Mamandur82]) gives the predicted value of the dependent variables based on study in [Mamandur82];
- A(max) is the maximum value of the dependent variables.

Figure 6.2 presents a subset of information in Table 6.2 graphically. The horizontal lines from bus number 3 to 6 give the minimum and maximum values of load bus voltages. This figure allows an examination of how the dependent variables are adjusted from the initial state to the final state for the techniques under consideration.

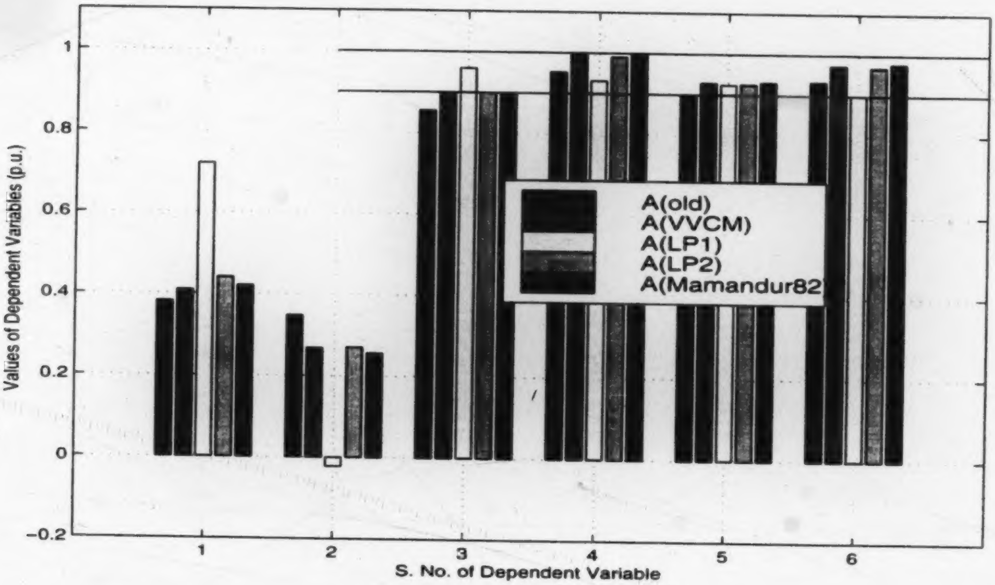


Figure 6.2: Dependent Variables (Case VVC1)

Table 6.3 compares the predicted values of the dependent variables for different techniques (VVCM, linear programming formulations 1 and 2) and [Mamandur82] solution with load flows to show the level of accuracy achieved by using sensitivities.

Type	A(VVCM)	A(LF-VV')	A(LP1)	A(LF-LP1)	A(LP2)	A(LF-LP2)	A(Mam)	A(LF-Mam)
Bus 1:Q	0.4082	0.4035	0.7209	0 .8563	0.4407	0.4360	0.4207	0.4323
Bus 2:Q	0.2663	0.2535	-0.0223	-0.0335	0.2697	0.2573	0.2564	0.2614
Bus 3:V	0.8989	0.9021	0.9598	0.9777	0.9002	0.9034	0.900	0.8980
Bus 4:V	0.9997	1.0037	0.9317	0.9182	0.9909	0.9954	0.999	0.9963
Bus 5:V	0.9289	0.9319	0.9248	0.9060	0.9261	0.9288	0.931	0.9235
Bus 6:V	0.9734	0.9786	0.9004	0.8789	0.9693	0.9745	0.978	0.9750

Table 6.3: Predicted vs Load Flow Values of Dependent Variables (Case VVC1)

A(Mam) stands for A[Mamandur82]).

It is significant to note that in [Mamandur82] formulation, the load bus voltage, in violation at bus number 3 could not be corrected. Figure 6.3 presents an error bar graph showing the error between the predicted values of the dependent variables for different techniques and load flows. This information has been derived from Table 6.3.

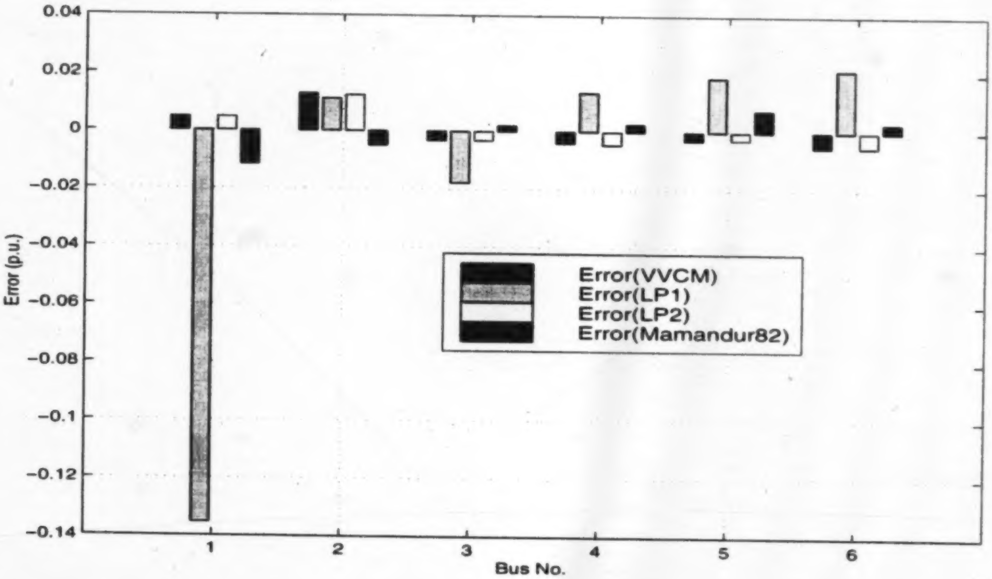


Figure 6.3: Error between Predicted and Load Flow Values of Dependent Variables (Case VVC1)

One notes, that errors are smaller for the VVCM, and LP2, as in these methods, the changes in the setting of the controllers are minimum. Method LP1 which causes a number of control variables to be disturbed has the maximum number of errors.

A linear programming method has the characteristic that the solution found by this method generally lies on the lower and upper limits of the variables [Rekliatis83]. This is evident by examining the solution by method LP1 in Table 6.1.

It is also significant to note that the errors produced by [Mamandur82] solution for the load bus voltages are *optimistic*, as all errors are positive as seen from Figure 6.3. On the other hand, the errors produced by the VVCM, LP1, LP2 solutions for load bus voltages are *pessimistic*.

2. **Case VVC2:** This case is for a 6 bus system. A comparison is made with results from [Mamandur82]. Method (LP2) in this thesis, resembles [Mamandur82] formulation. The system is at half-load level. The system has over-voltage violations at bus numbers 4 and 6 as seen from A(old) values in Table 6.5. To remedy this situation Table 6.4 presents recommended values of settings of the control variables for different techniques. NSTEPS(R) is computed from $\Sigma\Delta X$ assuming a controller step length of 0.01 p.u. for all type of controllers.

S.No.	Type	X(min)	X(old)	X(VVCM)	X(LP1)	X(LP2)	X([Mamandur82])	X(max)
1.	Tran 1: t	0.900	1.0250	1.0250	0.9000	1.0250	1.025	1.100
2.	Tran 2: t	0.900	1.1000	1.1000	0.9000	1.100	1.100	1.100
3.	Gen 1: V	1.000	1.0500	1.0200	1.0000	1.0172	1.016	1.100
4.	Gen 2 : V	1.000	1.1000	1.0900	1.0000	1.1000	1.100	1.150
5.	Q : 1	0.000	0.0000	0.0000	0.0000	0.0000	0.000	0.050
6.	Q : 4	0.000	0.0000	0.0000	0.0000	0.0000	0.000	0.055
	NSTEPS:			0.04	0.475	0.0328	0.034	
	NSTEPS(R):			4	48	4	4	

Table 6.4: Control Variables (Case VVC2)

Figure 6.4 presents a subset of information in Table 6.4 graphically. This figure enables an examination of how the controllers are adjusted by different techniques for remedying voltage/VAr violations.

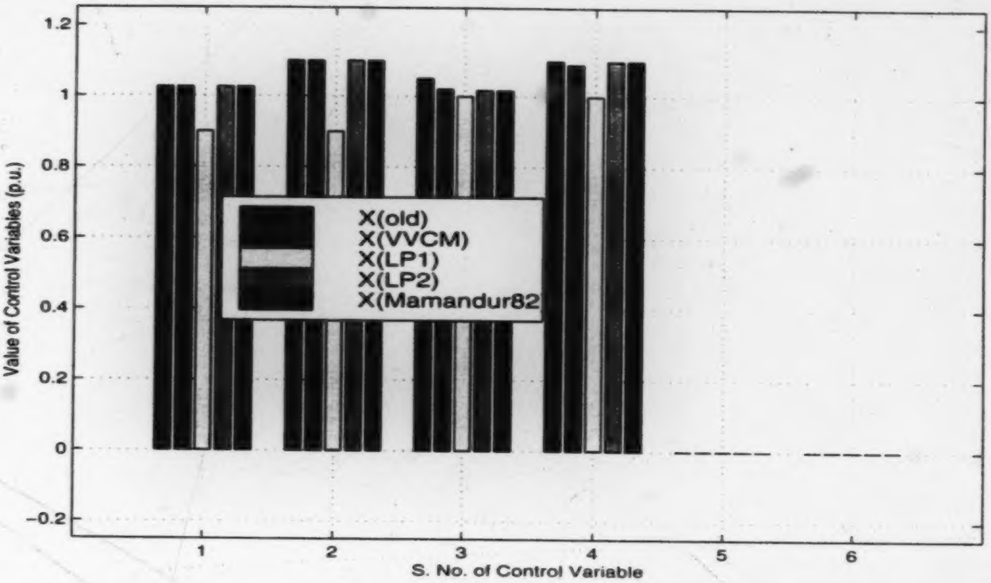


Figure 6.4: Control Variables (Case VVC2)

The NSTEPS(R) value for the VVCM, [Mamandur82] and LP2 methods are the same, that is, the number of step changes suggested by these techniques are the same. However, the time required to solve this problem by VVCM is less as compared to [Mamandur82] and LP2 methods. LP1 method gives a poor solution.

Table 6.5 gives the predicted values of the dependent variables for different techniques.

Type	A(min)	A(old)	A(VVCM)	A(LP1)	A(LP2)	A([Mamandur82])	A(max)
Bus 1: Q	-0.1955	0.0487	0.0308	0.2857	0.0146	0.0068	1.0004
Bus 2: Q	-0.1955	0.2163	0.2448	-0.0410	0.2621	0.2658	1.0004
Bus 3: V	0.9004	0.9350	0.9116	0.9862	0.9128	0.913	1.0004
Bus 4: V	0.9004	1.0270	1.0002	0.9300	1.0004	1.000	1.0004
Bus 5: V	0.9004	0.9962	0.9755	0.9774	0.9792	0.980	1.0004
Bus 6: V	0.9004	1.0173	0.9919	0.9132	0.9929	0.992	1.0004

Table 6.5: Dependent Variables (Case VVC2)

Figure 6.5 presents a subset of information in Table 6.5 graphically. The horizontal lines from bus number 3 to 6 give the minimum and maximum values of the load bus voltages. Buses 1 and 2 are the generator buses. This figure allows an examination of how the dependent variables are adjusted from the initial state to the final state for different techniques.

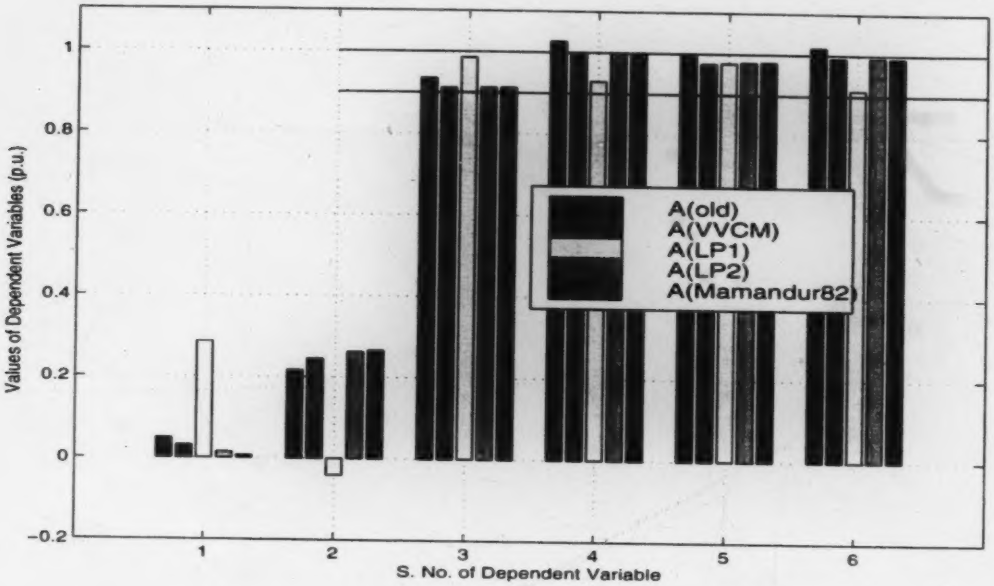


Figure 6.5: Dependent Variables (Case VVC2)

3. **Case VVC3:** This case is for a 6 bus system. A comparison is made with results from [Mamandur82]. In this thesis, method LP2 resembles [Mamandur82] formulation. The system is loaded at quarter of full-load level. The system has over-voltage violations at bus numbers 4, 5 and 6 as seen from A(old) values in Table 6.7. To remedy this situation Table 6.6 presents recommended values of settings of the control variables for different techniques:

S. No.	Type	X(min)	X(old)	X(VVCM)	X(LP1)	X(LP2)	X([Mamandur82])	X(max)
1.	Tran 1: t	0.900	1.0250	1.0250	0.9357	1.0250	1.025	1.100
2.	Tran 2: t	0.900	1.1000	1.1000	0.9374	1.1000	1.100	1.100
3.	Gen 1: V	1.000	1.0500	1.0000	1.0000	1.0000	1.000	1.100
4.	Gen 2: V	1.000	1.1000	1.0400	1.0000	1.0442	1.000	1.150
5.	Q:1(3)	0.000	0.0000	0.0000	0.0000	0.0000	0.000	0.050
6.	Q:2(6)	0.000	0.0000	0.0000	0.0000	0.0000	0.000	0.055
	NSTEPS:			0.11	0.4019	0.1058	0.15	
	NSTEPS(R):			11	41	11	15	

Table 6.6: Control Variables (Case VVC3)

Figure 6.6 presents a subset of information in Table 6.6 graphically. This figure enables an examination of how the controllers are adjusted by different techniques for remedying voltage/VAr violations.

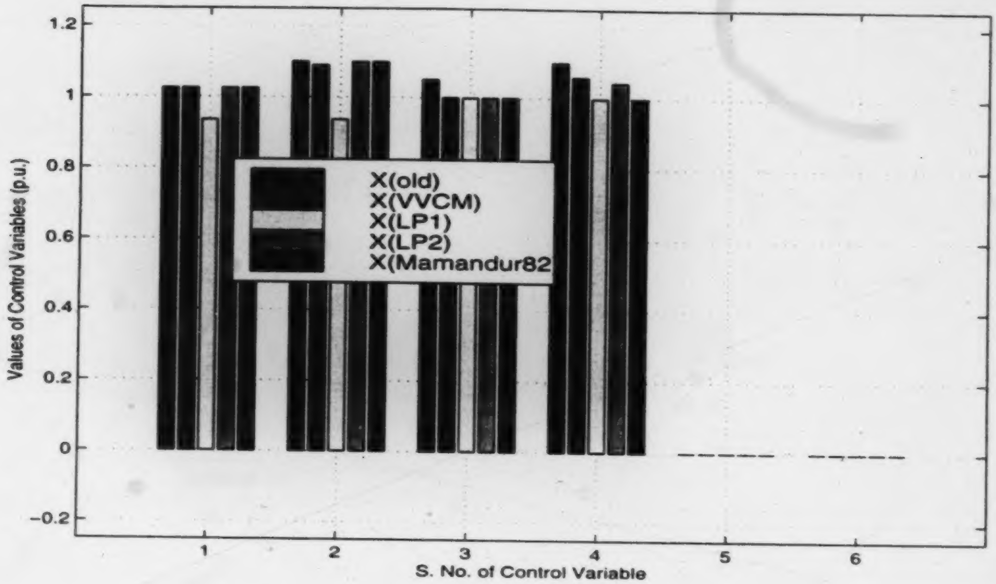


Figure 6.6: Control Variables (Case VVC3)

One notes, that the VVCM provides the best solution, based on the criteria of minimizing the number of step changes in controller settings, that is the value of NSTEPS(R) for the VVCM. This is followed by LP2, [Mamandur82] and LP1.

Table 6.7 gives the predicted values of the dependent variables for different techniques.

Type	A(min)	A(old)	A(VVCM)	A(LP1)	A(LP2)	A([Mamandur82])	A(max)
Bus 1: Q	-0.1955	-0.0603	-0.0322	0.1372	-0.0379	0.0006	1.0004
Bus 2: Q	-0.1955	0.1806	0.1704	-0.0375	0.1760	0.0839	1.0004
Bus 3: V	0.9004	0.9662	0.9137	1.0004	0.9150	0.906	1.0004
Bus 4: V	0.9004	1.0554	0.9995	0.9659	1.0006	0.994	1.0004
Bus 5: V	0.9004	1.0347	0.9775	1.0004	0.9797	0.961	1.0004
Bus 6: V	0.9004	1.0498	0.9939	0.9552	0.9952	0.986	1.0004

Table 6.7: Dependent Variables (Case VVC3)

Figure 6.7 presents a subset of information in Table 6.7 graphically. The horizontal lines from bus number 3 to 6 give the minimum and maximum values of load bus voltages. Buses 1 and 2 are the generator buses. This figure allows an examination of how the dependent variables are adjusted from the initial state to the final state for different techniques.

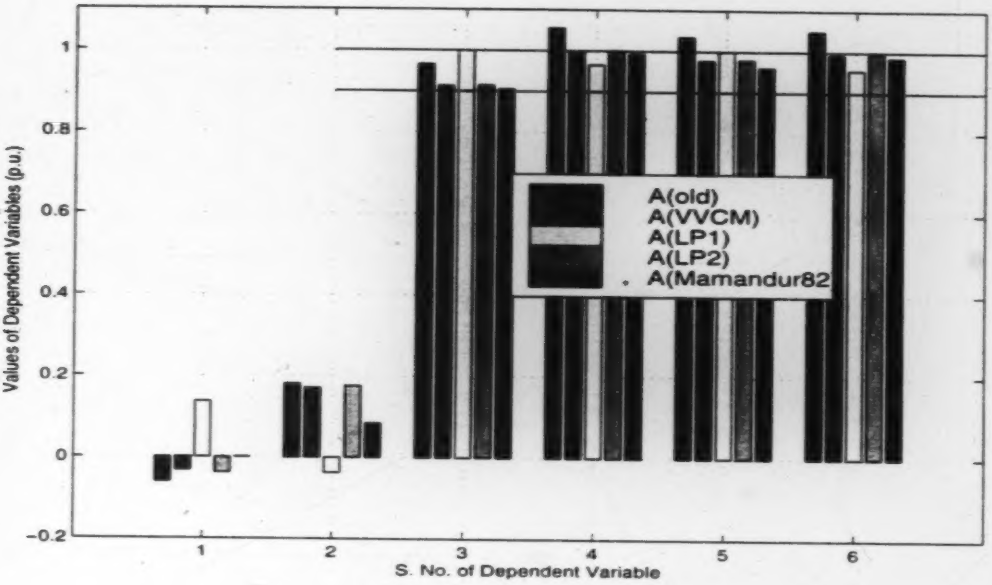


Figure 6.7: Dependent Variables (Case VVC3)

4. **Case 4:** This case is for IEEE 14 bus system. A comparison is made with results from [Singh93]. It seems that there is some typographical error in [Singh93]. This paper suggests that the system be loaded upto 4.2 times the full load. At this load level, the load flow does not converge. Hence, the system is loaded at 3.2 times the full-load level.

   Comment: *Loading of 3.2 times the full-load, does not reflect the actual power system operational scenario. At these load levels, relays would trip in order to save the transmission lines and transformers. In these studies, it is just a means to artificially stress the system to induce voltage/VAr violations. This is an alternative method for inducing voltage/VAr violations in place of simulating line and generator contingencies. In this author's opinion, a possible advantage of using this way of stressing the system is that sensitivities need not be recalculated as the topology of the connected system does not change. On the other hand, with a line and/or generator contingency,* *the system topology changes which necessitates recalculation of sensitivities.*

The system has an under-voltage violation at bus numbers 6, 9, 10 and 14 as seen from A(old) values in Table 6.9. To remedy this situation Table 6.8 presents recommended values of settings of control variables for different techniques. NSTEPS(R) is computed from $\Sigma\Delta X$ assuming that the step length of a transformer tap is 0.02, for generator voltage is 0.01, and for a capacitor is 0.1 p.u..

S. No.	Type	X(min)	X(old)	X(VVCM)	X(LP1)	X(LP2)	X(max)
1.	Tran 1: t	0.9000	1.0000	1.0000	0.9000	1.1000	1.1000
2.	Tran 2: t	0.9000	1.0000	1.0000	1.0791	1.1000	1.1000
3.	Tran 3: t	0.9000	1.0000	1.0000	1.0354	0.9975	1.1000
4.	Gen 1: V	0.9000	1.0600	1.0600	1.1000	1.0589	1.1000
5.	Gen 2: V	0.9000	1.0450	1.0450	1.0903	1.0531	1.1000
6.	Gen 3: V	0.9000	1.0100	1.0100	1.0784	1.0458	1.1000
7.	Gen 4: V	0.9000	1.0700	1.1000	1.1000	1.1000	1.1000
8.	Gen 5: V	0.9000	1.0900	1.0900	1.1000	1.1000	1.1000
9.	Q:1(9)	0.0000	0.0000	0.5000	0.0732	0.1609	0.5000
	$\Sigma\Delta X$			0.53	0.4814	0.4126	
	NSTEPS(R):			8	43	26	

Table 6.8: Control Variables (Case VVC4)

For a similar loading, [Singh93] reports that voltages at buses 9, 10 and 14 are 0.87535, 0.85776 and 0.80876 p.u. respectively. [Singh93] solution involves controller changes ($\Sigma\Delta X$) of 0.63 p.u. and number of step changes (NSTEPS(R)) equal to 25.

One notes that the VVCM performs better than the results reported in [Singh93] for more number of voltage violations. [Singh93] solution incorporates change in setting of two transformers with taps (total tap movement is 0.22 p.u.). In contrast to the [Singh93] solution, in the VVCM solution, no transformer tap action is suggested. One possible reason for this could be the use of more accurate model for tap changing transformer, a method developed in this thesis.

Figure 6.8 presents a subset of information in Table 6.8 graphically. This figure enables an examination of how the controllers are adjusted by different techniques for remedying voltage/VAr violations.

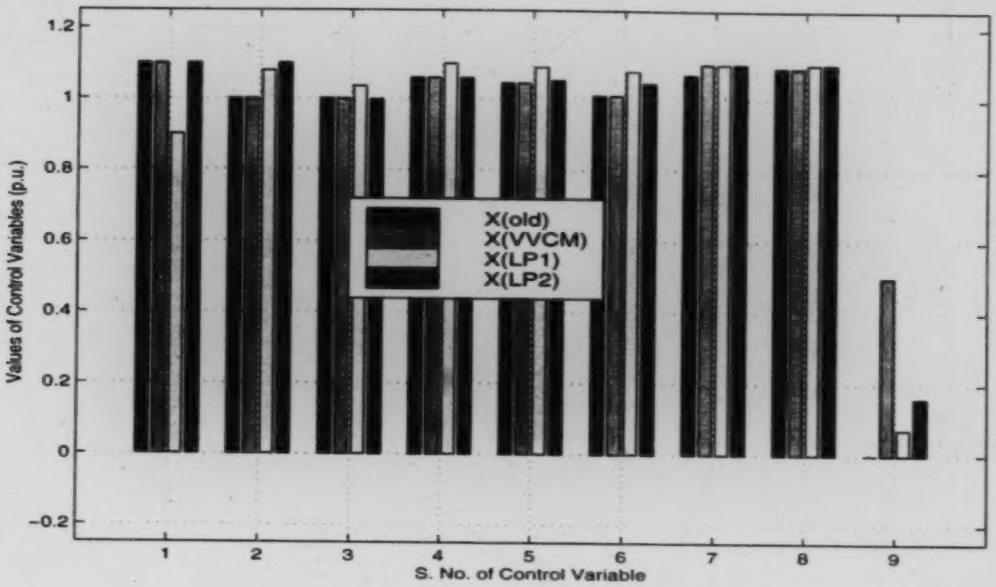


Figure 6.8: Control Variables (Case VVC4)

Table 6.9 gives the predicted values of the dependent variables for different techniques applied to this IEEE 14 bus system.

Type	A(min)	A(old)	A(VVCM)	A(LP1)	A(LP2)	A(max)
Bus 1: Q	-2.9950	0.3969	0.3356	0.2918	0.1996	5.2500
Bus 2: Q	-2.9950	5.2368	5.0710	5.2500	5.2500	5.2500
Bus 3: Q	-2.9950	2.7553	2.6674	3.0309	3.0481	5.2500
Bus 4: Q	-2.9950	2.3917	2.2769	2.1984	2.3086	5.2500
Bus 5: Q	-2.9950	0.9966	0.7460	0.6239	0.6688	5.2500
Bus 6: V	0.8950	0.8893	0.9053	0.9270	0.8958	1.1040
Bus 7: V	0.8950	0.8947	0.9080	0.9380	0.9028	1.1040
Bus 8: V	0.8950	0.9289	0.9694	0.9992	0.9919	1.1040
Bus 9: V	0.8950	0.8666	0.9453	0.9406	0.9414	1.1040
Bus 10: V	0.8950	0.8741	0.9444	0.9405	0.9412	1.1040
Bus 11: V	0.8950	0.9580	1.0087	1.0067	1.0070	1.1040
Bus 12: V	0.8950	1.0054	1.0399	1.0396	1.0396	1.1040
Bus 13: V	0.8950	0.9753	1.0133	1.0126	1.0127	1.1040
Bus 14: V	0.8950	0.8370	0.8979	0.8950	0.8955	1.1040

Table 6.9: Dependent Variables (Case VVC4)

Figure 6.9 presents a subset of information in Table 6.9 graphically. The horizontal lines from bus number 6 to 14 give the minimum and maximum values of load bus voltages. Buses 1 to 5 are the generator buses. This figure allows an examination of how the dependent variables are adjusted from the initial state to the final state for different techniques.

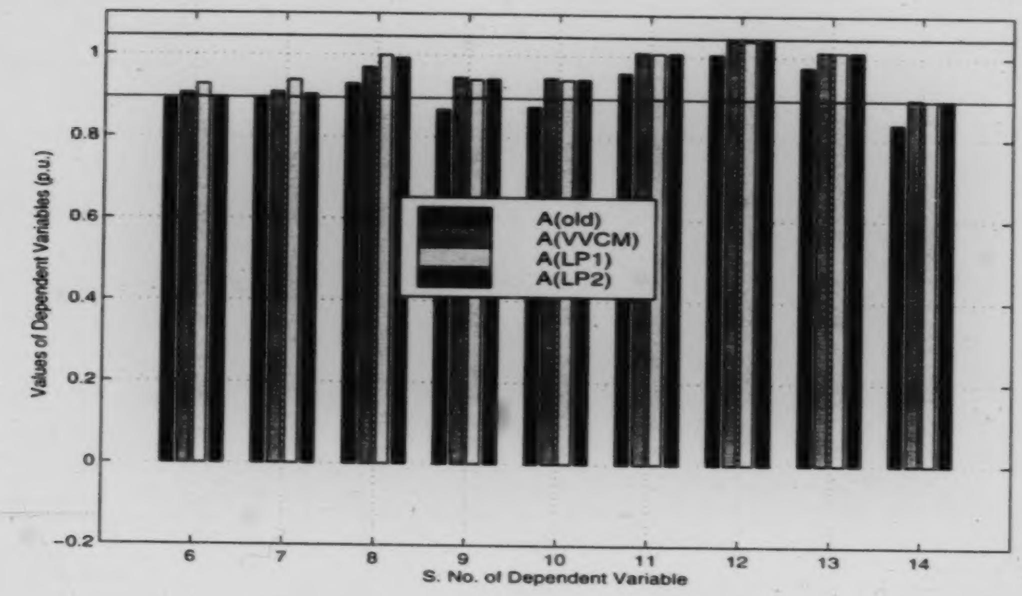


Figure 6.9: Dependent Variables (Case VVC4)

Table 6.10 compares the predicted values of the dependent variables for different techniques (the VVCM and linear programming formulations 1 and 2) with load flows.

Type	A(VVCM)	A(LF-VVCM)	A(LP1)	A(LF-LP1)	A(LP2)	A(LF-LP2)
Bus 1: Q	0.3356	0.2891	0.2918	-0.0385	0.1996	0.1580
Bus 2: Q	5.0710	4.9200	5.2500	4.8650	5.2500	5.1864
Bus 3: Q	2.6674	2.6082	3.0309	2.9011	3.0481	3.0661
Bus 4: Q	2.2769	2.2299	2.1984	2.1695	2.3086	2.2739
Bus 5: Q	0.7460	0.7162	0.6239	0.5689	0.6688	0.6157
Bus 6: V	0.9053	0.9102	0.9270	0.9439	0.8958	0.9006
Bus 7: V	0.9080	0.9141	0.9380	0.9565	0.9028	0.9086
Bus 8: V	0.9694	0.9743	0.9992	1.0089	0.9919	0.9905
Bus 9: V	0.9453	0.9511	0.9406	0.9546	0.9414	0.9444
Bus 10: V	0.9444	0.9512	0.9405	0.9512	0.9412	0.9454
Bus 11: V	1.0087	1.0126	1.0067	1.0142	1.0070	1.0096
Bus 12: V	1.0399	1.0417	1.0396	1.0420	1.0396	1.0412
Bus 13: V	1.0133	1.0168	1.0126	1.0172	1.0127	1.0155
Bus 14: V	0.8979	0.9093	0.8950	0.9117	0.8955	0.9046

Table 6.10: Predicted vs Load Flow Values of Dependent Variables (Case VVC4)

Figure 6.10 presents an error bar graph showing the error between the predicted values of the dependent variables for three techniques and load flows. This information has been derived from Table 6.10.

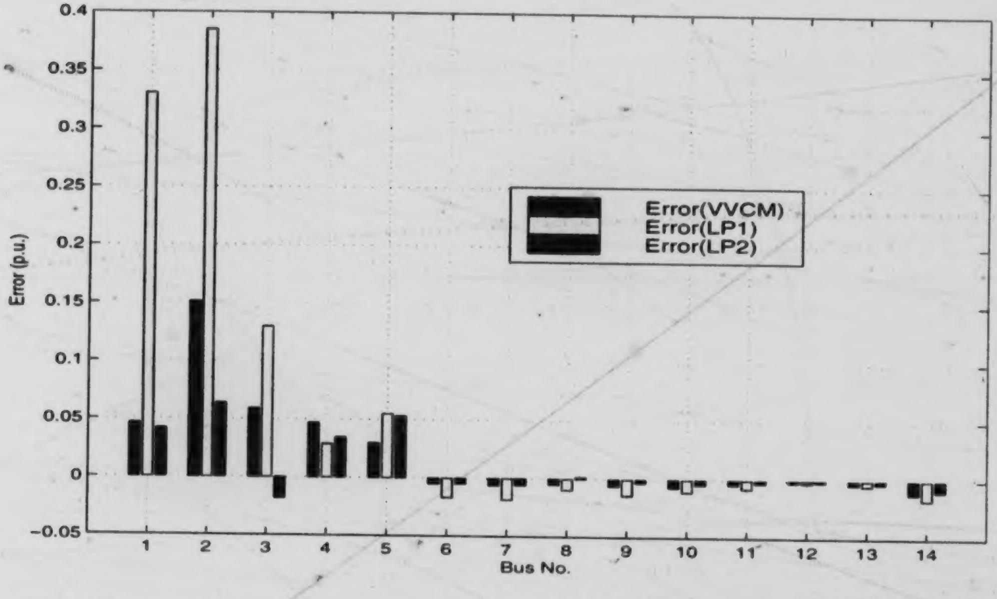


Figure 6.10: Error between Predicted and Load Flow Values of Dependent Variables (Case VVC4)

One notes, that errors are least for the (LP2), followed by (VVCM), as in these methods the changes in the setting of the controllers is minimum. Method (LP1) which causes a number of control variables to be disturbed has maximum number of errors.

5. **Case VVC5:** This case is for IEEE 30 bus system. The system is loaded at full-load level. The system has under-voltage violations at bus numbers 26, 29 and 30 as seen from A(old) values in Table 6.12. To remedy this situation Table 6.11 presents recommended values of settings of the control variables for different techniques. NSTEPS(R) is computed from $\Sigma\Delta X$ assuming controller step lengths of 0.01 p.u. for all types of controllers.

S. No.	Type	X(min)	X(old)	X(VVCM)	X(LP1)	X(LP2)	X(max)
1.	Tran 1: t	0.9000	0.9780	0.9780	0.9048	0.9780	1.1000
2.	Tran 2: t	0.9000	0.9690	0.9690	0.9000	0.9690	1.1000
3.	Tran 3: t	0.9000	0.9320	0.9320	0.9000	0.9320	1.1000
4.	Tran 4: t	0.9000	0.9680	1.0080	0.9896	0.9985	1.1000
5.	Gen 1: V	0.9000	1.0600	1.0600	0.9000	1.0600	1.1000
6.	Gen 2: V	0.9000	1.0450	1.0450	0.9000	1.0450	1.1000
7.	Gen 3: V	0.9000	1.0100	1.0100	0.9000	1.0100	1.1000
8.	Gen 4: V	0.9000	1.0100	1.1000	1.1000	1.1000	1.1000
9.	Gen 5: V	0.9000	1.0820	1.0820	1.1000	1.0820	1.1000
10.	Gen 6: V	0.9000	1.0710	1.0710	1.1000	1.0710	1.1000
11.	Q:1(9)	0.0000	0.0000	0.0000	0.00	0.00	0.0000
12.	Q:2(24)	0.0000	0.0000	0.0000	0.0214	0.00	0.0430
	$\Sigma\Delta X$:			0.1380	0.7478	0.1205	
	NSTEPS(R):			14	75	13	

Table 6.11: Control Variables (Case VVC5)

Methods VVCM and LP2 give comparable results, based on the criteria of number of step changes of controllers, that is, NSTEPS(R). However, the time required by the VVCM is less as compared to the LP2 method.

Figure 6.11 presents a subset of information in Table 6.11 graphically. This figure enables an examination of how the controllers are adjusted by different techniques for remedying voltage/VAr violations.

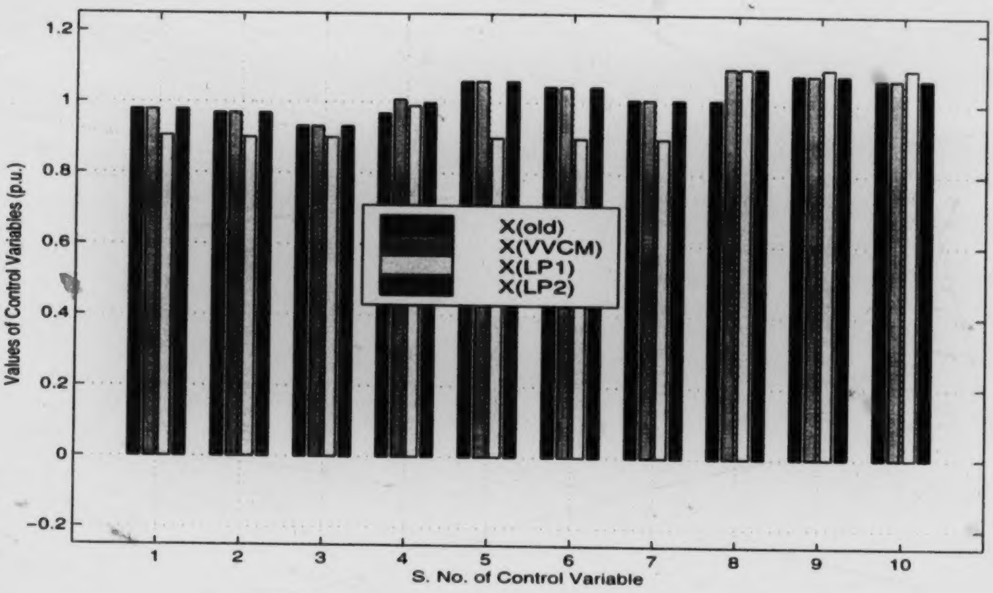


Figure 6.11: Control Variables (Case VVC5)

Table 6.12 gives the predicted values of the dependent variables for different techniques.

Type	A(min)	A(old)	A(VVCM)	A(LP1)	A(LP2)	A(max)
Bus 1: Q	-2.9950	-0.2258	-0.2261	-1.0739	-0.2261	2.9950
Bus 2: Q	-2.9950	0.5305	0.5300	-0.9047	0.5301	2.9950
Bus 3: Q	-2.9950	0.3587	0.3589	-0.0365	0.3558	2.9950
Bus 4: Q	-2.9950	0.2850	0.2931	2.4385	0.2910	2.9950
Bus 5: Q	-2.9950	0.3822	0.3743	0.6073	0.3672	2.9950
Bus 6: Q	0.9450	0.5225	0.5109	0.8357	0.5137	1.1040
Bus 7: V	0.9450	1.0298	1.0299	0.9761	1.0299	1.1040
Bus 8: V	0.9450	1.0234	1.0235	0.9921	1.0235	1.1040
Bus 9: V	0.9450	1.0140	1.0139	1.0241	1.0139	1.1040
Bus 10: V	0.9450	1.0042	1.0041	0.9651	1.0041	1.1040
Bus 11: V	0.9450	1.0085	1.0100	0.9833	1.0097	1.1040
Bus 12: V	0.9450	0.9803	0.9835	0.9560	0.9827	1.1040
Bus 13: V	0.9450	1.0027	1.0042	0.9908	1.0039	1.1040
Bus 14: V	0.9450	0.9855	0.9878	0.9729	0.9872	1.1040
Bus 15: V	0.9450	0.9794	0.9824	0.9665	0.9817	1.1040
Bus 16: V	0.9450	0.9856	0.9878	0.9685	0.9873	1.1040
Bus 17: V	0.9450	0.9763	0.9791	0.9541	0.9785	1.1040
Bus 18: V	0.9450	0.9687	0.9718	0.9515	0.9711	1.1040
Bus 19: V	0.9450	0.9658	0.9689	0.9459	0.9682	1.1040
Bus 20: V	0.9450	0.9699	0.9731	0.9487	0.9724	1.1040
Bus 21: V	0.9450	0.9662	0.9705	0.9450	0.9995	1.1040
Bus 22: V	0.9450	0.9666	0.9713	0.9465	0.9702	1.1040
Bus 23: V	0.9450	0.9634	0.9691	0.9535	0.9677	1.1040
Bus 24: V	0.9450	0.9511	0.9602	0.9450	0.9581	1.1040
Bus 25: V	0.9450	0.9487	0.9702	0.9640	0.9651	1.1040
Bus 26: V	0.9450	0.9297	0.9512	0.9450	0.9461	1.1040
Bus 27: V	0.9450	0.9565	0.9856	0.9850	0.9787	1.1040
Bus 28: V	0.9450	1.0087	1.0073	1.0348	1.0076	1.1040
Bus 29: V	0.9450	0.9351	0.9642	0.9636	0.9573	1.1040
Bus 30: V	0.9450	0.9227	0.9519	0.9512	0.9450	1.1040

Table 6.12: Dependent Variables (Case VVC5)

Figures 6.12 and 6.13 present a subset of information in Table 6.12 graphically. The horizontal lines from bus number 7 to 30 give the minimum and maximum values of load bus voltages. Buses 1 to 6 are the generator buses. This figure allows an examination of how the dependent variables are adjusted from the initial state to the final state by different techniques.

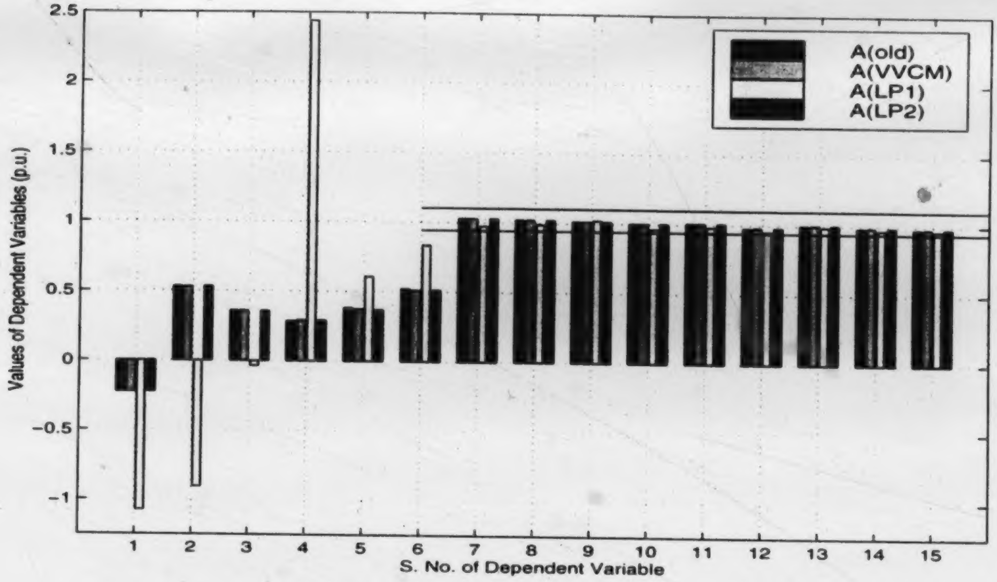


Figure 6.12: Dependent Variables, Buses 1-15 (Case VVC5)

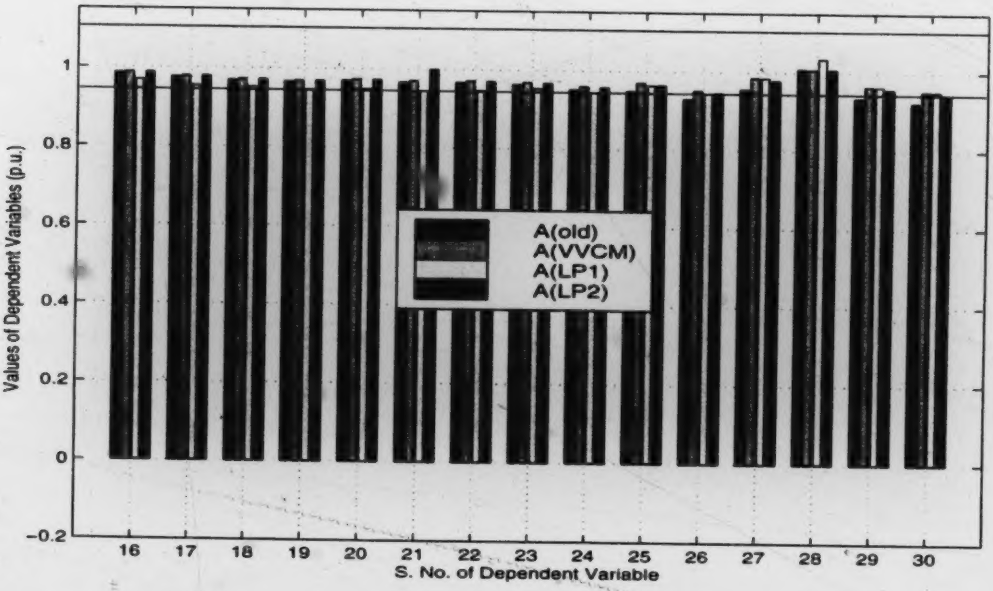


Figure 6.13: Dependent Variables, Buses 16-30 (Case VVC5)

Table 6.13 compares the predicted values of dependent variables for different techniques (the VVCM and linear programming formulations 1 and 2) with load flows.

Type	A(VVCM)	A(LF-VVCM)	A(LP1)	A(LF-LP1)	A(LP2)	A(LF-LP2)
Bus 1: Q	-0.2258	-0.2261	-1.0739	-0.9433	-0.2261	-0.2260
Bus 2: Q	0.5305	0.5310	-0.9047	-0.5733	0.5301	0.5309
Bus 3: Q	0.3587	0.3596	-0.0365	0.0330	0.3588	0.3594
Bus 4: Q	0.2850	0.2973	2.4385	2.8663	0.2912	0.2943
Bus 5: Q	0.3822	0.3759	0.6073	0.6393	0.3762	0.3773
Bus 6: Q	0.5225	0.5133	0.8357	0.8747	0.5137	0.5154
Bus 7: V	1.0298	1.0299	0.9761	0.9683	1.0299	1.0299
Bus 8: V	1.0234	1.0235	0.9921	0.9849	1.0235	1.0234
Bus 9: V	1.0140	1.0138	1.0241	1.0198	1.0139	1.0139
Bus 10: V	1.0042	1.0041	0.9651	0.9620	1.0041	1.0041
Bus 11: V	1.0085	1.0098	0.9833	0.9791	1.0097	1.0095
Bus 12: V	0.9803	0.9829	0.9560	0.9527	0.9827	0.9823
Bus 13: V	1.0027	1.0039	0.9908	0.9887	1.0039	1.0036
Bus 14: V	0.9855	0.9874	0.9729	0.9709	0.9872	0.9869
Bus 15: V	0.9794	0.9819	0.9665	0.9640	0.9817	0.9813
Bus 16: V	0.9856	0.9874	0.9685	0.9657	0.9873	0.9870
Bus 17: V	0.9763	0.9713	0.9541	0.9508	0.9785	0.9781
Bus 18: V	0.9687	0.9684	0.9515	0.9486	0.9711	0.9707
Bus 19: V	0.9658	0.9725	0.9459	0.9430	0.9682	0.9678
Bus 20: V	0.9699	0.9698	0.9487	0.9458	0.9724	0.9719
Bus 21: V	0.9662	0.9698	0.9450	0.9415	0.9695	0.9690
Bus 22: V	0.9666	0.9706	0.9465	0.9430	0.9702	0.9697
Bus 23: V	0.9634	0.9684	0.9535	0.9505	0.9677	0.9672
Bus 24: V	0.9511	0.9592	0.9450	0.9406	0.9581	0.9573
Bus 25: V	0.9487	0.9679	0.9640	0.9585	0.9651	0.9635
Bus 26: V	0.9297	0.9493	0.9450	0.9397	0.9461	0.9448
Bus 27: V	0.9565	0.9824	0.9850	0.9788	0.9787	0.9764
Bus 28: V	1.0087	1.0070	1.03848	1.0312	1.0076	1.0074
Bus 29: V	0.9351	0.9616	0.9636	0.9580	0.9573	0.9555
Bus 30: V	0.9227	0.9496	0.9512	0.9459	0.9450	0.9434

Table 6.13: Predicted vs Load Flow Values of Dependent Variables (Case VVC5)

Figure 6.14 presents an error bar graph showing the error between the predicted values of the dependent variables for different techniques and load flows. This information has been derived from Table 6.13.

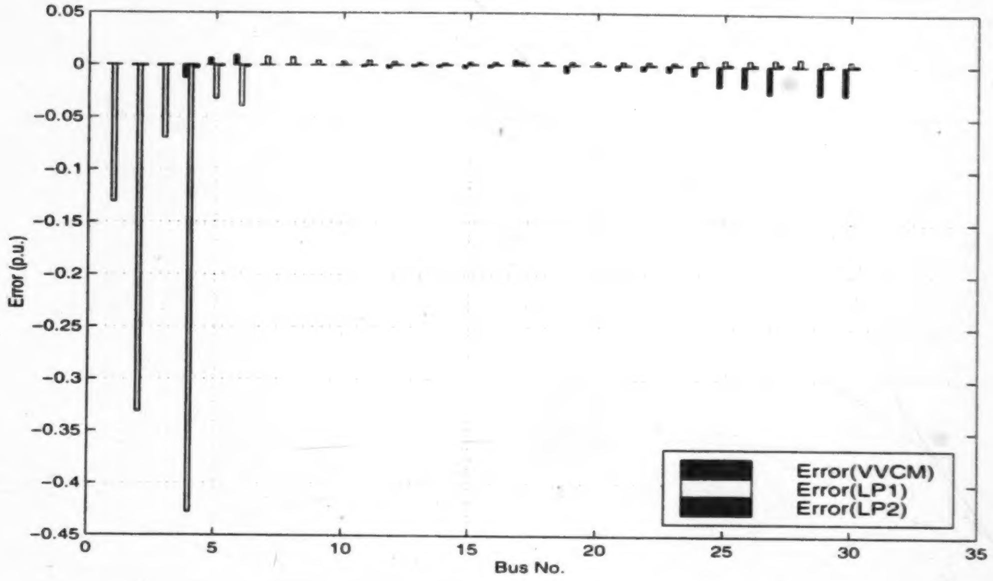


Figure 6.14: Predicted vs Load Flow Values of Dependent Variables (Case 5)

Errors are maximum with LP(1) method, as in this method maximum number of control variables are changed.

6. **Case VVC6:** This case is for IEEE 30 bus system. A comparison is made with results from [Singh93]. The system is loaded at 2.1 times[2] the full-load level. The system has under-voltage violations at bus numbers 14-27 and 29-30 as seen from A(old) values in Table 6.15. To remedy this situation Table 6.14 presents recommended values of settings of the control variables for different techniques. NSTEPS(R) is computed from $\Sigma\Delta X$ assuming that the step length of a transformer tap is 0.02, generator voltage is 0.01, and capacitor/reactor is 0.1 p.u..

---

[2] Loading of 2.1 times the full-load, does not reflect the actual power system operational scenario. At these load levels, relays would trip in order to save the transmission lines and transformers. In these studies, it is just a means to artificially stress the system to induce voltage/VAr violations. This is an alternative method for inducing voltage/VAr violations in place of simulating line and generator contingencies. In this author's opinion, a possible advantage of using this way of stressing the system is that sensitivities need not be recalculated as the topology of the connected system does not change. On the other hand, with a line and/or generator contingency, the system topology changes, which necessitates recalculation of sensitivities.

S. No.	Type	X(min)	X(old)	X(VVCM)	X(LP1)	X(LP2)	X(max)
1.	Tran 1: t	0.9000	1.0000	1.0000	1.0231	1.0000	1.1000
2.	Tran 2: t	0.9000	1.0000	1.0000	1.0217	1.0000	1.1000
3.	Tran 3: t	0.9000	1.0000	1.0000	1.0201	1.0000	1.1000
4.	Tran 4: t	0.9000	1.0000	1.1000	1.1000	1.1000	1.1000
5.	Gen 1: V	0.9000	1.0600	1.0600	1.0872	1.0600	1.1000
6.	Gen 2: V	0.9000	1.0450	1.0450	1.1000	1.0864	1.1000
7.	Gen 3: V	0.9000	1.0100	1.0100	1.0478	1.0100	1.1000
8.	Gen 4: V	0.9000	1.0700	1.1000	1.1000	1.1000	1.1000
9.	Gen 5: V	0.9000	1.0820	1.0820	1.1000	1.1000	1.1000
10.	Gen 6: V	0.9000	1.0710	1.0710	1.1000	1.1000	1.1000
11.	Q:1(9)	0.0000	0.0000	0.0000	0.0000	0.0000	0.1900
12.	Q:2(24)	0.0000	0.0000	0.4000	0.0627	0.1301	0.430
	$\Sigma\Delta X$:			0.53	0.411	0.3485	
	NSTEPS(R):			12	33	18	

Table 6.14: Control Variables (Case VVC6)

For a similar loading, [Singh93] reports that voltages at buses 25, 26, 29 and 30 are 0.94649, 0.90541, 0.90311 and 0.87505 p.u. respectively. [Singh93] solution involves controller changes ($\Sigma\Delta X$) of 0.57 p.u. and number of step changes (NSTEPS(R)) equal to 21.

One notes that the VVCM method performs better than the results reported in [Singh93] for more number and more serious voltage violations. [Singh93]'s solution has no change in any of the transformer tap settings. In contrast, in the VVCM solution, one transformer tap action is suggested. One reason for this could be the use of more accurate model for tap changing transformer, a method developed in this thesis. The VVCM also performs better than the LP2 method.

Figure 6.15 presents a subset of information in Table 6.14 graphically. This figure enables an examination of how the controller are adjusted by different techniques for remedying voltage/VAr violations.

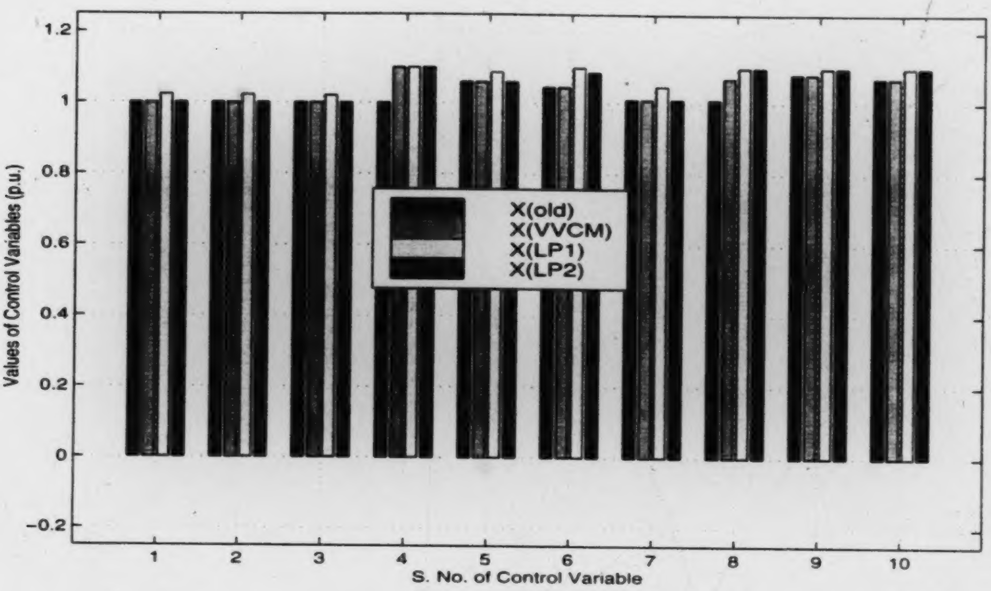


Figure 6.15: Control Variables (Case VVC6)

Table 6.15 gives the predicted values of the dependent variables for different techniques.

Type	A(min)	A(old)	A(VVCM)	A(LP1)	A(LP2)	A(max)
Bus 1: Q	-2.9950	-0.3790	-0.5112	-1.0119	-1.3401	2.9950
Bus 2: Q	-2.9950	2.2564	1.8661	2.7157	2.9948	2.9950
Bus 3: Q	-2.9950	1.3777	1.1866	1.1391	0.8684	2.9950
Bus 4: Q	-2.9950	1.7257	2.2644	2.3069	2.4496	2.9950
Bus 5: Q	-2.9950	0.5511	0.3226	0.2932	0.3550	2.9950
Bus 6: Q	0.9450	0.7262	0.4551	0.4799	0.5484	1.1040
Bus 7: V	0.9450	0.9819	1.0059	1.0358	1.0248	1.1040
Bus 8: V	0.9450	0.9740	1.0030	1.0334	1.0259	1.1040
Bus 9: V	0.9450	0.9803	1.0199	1.0489	1.0435	1.1040
Bus 10: V	0.9450	0.9729	0.9964	1.0292	1.0103	1.1040
Bus 11: V	0.9450	0.9761	1.0200	1.0436	1.0318	1.1040
Bus 12: V	0.9450	0.9233	0.9927	1.0047	0.9949	1.1040
Bus 13: V	0.9450	0.9748	1.0115	1.0373	1.0283	1.1040
Bus 14: V	0.9450	0.9360	0.9852	1.0050	0.9968	1.1040
Bus 15: V	0.9450	0.9221	0.9825	0.9954	0.9882	1.1040
Bus 16: V	0.9450	0.9391	0.9897	1.0096	1.0003	1.1040
Bus 17: V	0.9450	0.9198	0.9835	0.9979	0.9883	1.1040
Bus 18: V	0.9450	0.8974	0.9619	0.9746	0.9664	1.1040
Bus 19: V	0.9450	0.8905	0.9574	0.9700	0.9611	1.1040
Bus 20: V	0.9450	0.8999	0.9681	0.9806	0.9714	1.1040
Bus 21: V	0.9450	0.8919	0.9765	0.9781	0.9707	1.1040
Bus 22: V	0.9450	0.8930	0.9825	0.9805	0.9740	1.1040
Bus 23: V	0.9450	0.8868	0.9857	0.9732	0.9708	1.1040
Bus 24: V	0.9450	0.8601	1.0104	0.9640	0.9679	1.1040
Bus 25: V	0.9450	0.8637	1.0099	0.9905	0.9915	1.1040
Bus 26: V	0.9450	0.8181	0.9644	0.9450	0.9460	1.1040
Bus 27: V	0.9450	0.8883	1.0318	1.0292	1.0284	1.11490
Bus 28: V	0.9450	0.9711	1.0167	1.0426	1.0388	1.1040
Bus 29: V	0.9450	0.8349	0.9784	0.9758	0.9750	1.1040
Bus 30: V	0.9450	0.8041	0.9476	0.9450	0.9442	1.1040

Table 6.15: Dependent Variables (Case VVC6)

Figures 6.16 and 6.17 present a subset of information in Table 6.15 graphically. The horizontal lines from bus number 7 to 30 give the minimum and maximum values of load bus voltages. Buses 1 to 6 are the generator buses. These figures allow an examination of how the dependent variables are adjusted from the initial state to the final state for different techniques.

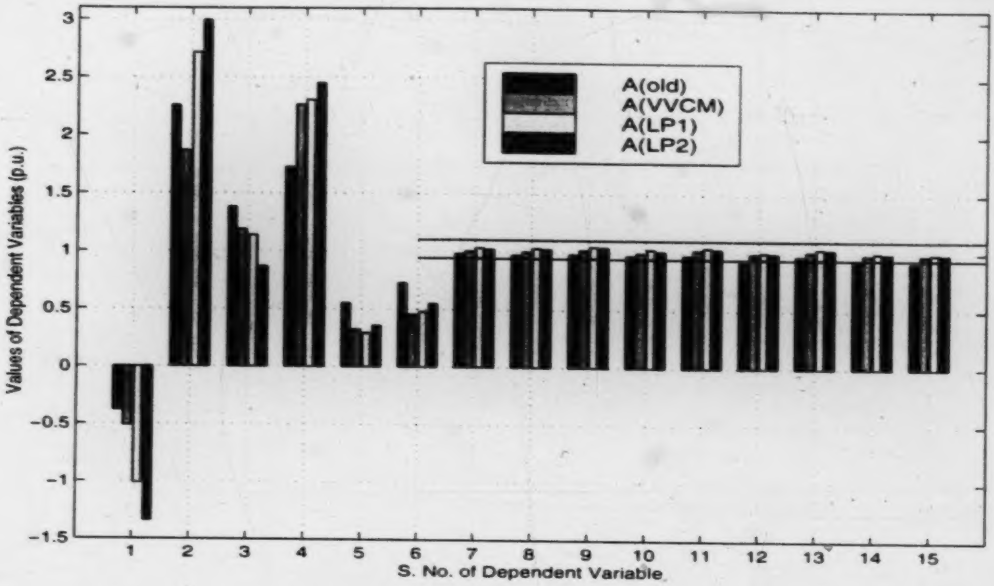


Figure 6.16: Dependent Variables, Buses 1-15 (Case VVC6)

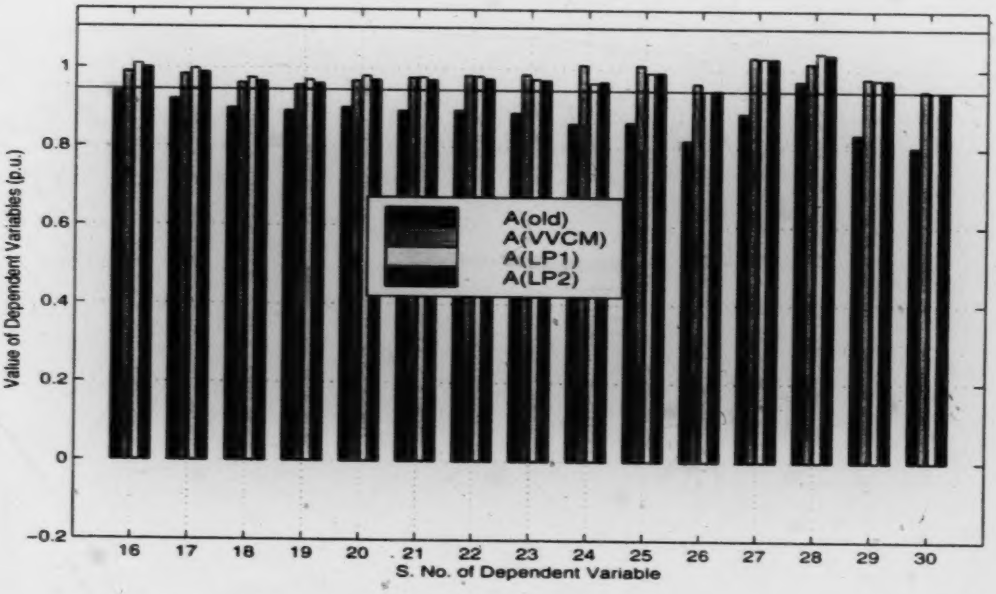


Figure 6.17: Dependent Variables, Buses 16-30 (Case VVC6)

7. **Case VVC7:** This case is for IEEE 30 bus system. This case has been adapted from [Cheng89]. The system is loaded at full-load level. Line 27-30 is switched

off. The system has under-voltage violations at bus numbers 24-26 and 29-30 as seen from A(old) values in Table 6.17. To remedy this situation Table 6.16 presents recommended values of the settings of the control variables for different techniques. NSTEPS(R) is computed from $\Sigma\Delta X$ assuming that the step length of all controllers is 0.01 p.u..

S. No.	Type	X(min)	X(old)	X(VVCM)	X(LP1)	X(LP2)	X(max)
1.	Tran 1: t	0.9000	0.9524	0.9524	0.9588	0.9524	1.1000
2.	Tran 2: t	0.9000	0.9346	0.9346	0.9415	0.9346	1.1000
3.	Tran 3: t	0.9000	0.9524	0.9524	0.9595	0.9524	1.1000
4.	Tran 4: t	0.9000	0.9615	1.0915	1.0385	1.0925	1.1000
5.	Gen 1: V	0.9000	1.0600	1.0600	1.0594	1.0600	1.1000
6.	Gen 2: V	0.9000	1.0450	1.0450	1.0563	1.0450	1.1000
7.	Gen 3: V	0.9000	1.0100	1.0100	1.0167	1.0100	1.1000
8.	Gen 4: V	0.9000	1.0100	1.0200	1.0646	1.0100	1.1000
9.	Gen 5: V	0.9000	1.0820	1.0820	1.0934	1.0820	1.1000
10.	Gen 6: V	0.9000	1.0710	1.0710	1.0857	1.0710	1.1000
11.	Q:1(9)	0.0000	0.0000	0.0000	0.0008	0.0000	0.0000
12.	Q:2(24)	0.0000	0.0000	0.0000	0.0114	0.0000	0.4000
	$\Sigma\Delta X$:			0.14	0.2089	0.1310	
	NSTEPS(R):			14	15	14	

Table 6.16: Control Variables (Case VVC7)

The VVCM performs same as method LP2 based on the NSTEPS(R) criteria. However, the time requirements for the VVCM is less as compared to LP2. Method LP1 although has a comparable solution to the VVCM is poor in the sense that it disturbs a number of control variables.

Figure 6.18 presents a subset of information in Table 6.16 graphically. This figure enables an examination of how the controllers are adjusted by different techniques for remedying voltage/VAr violations.

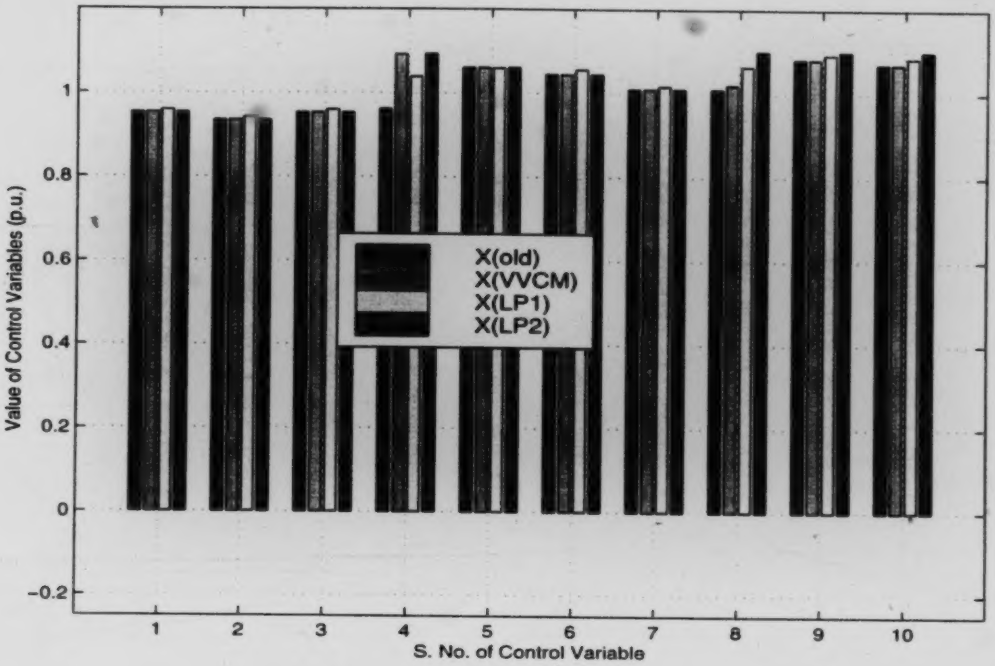


Figure 6.18: Control Variables (Case VVC7)

Table 6.17 gives the predicted values of the dependent variables for different techniques.

Type	A(min)	A(old)	A(VVCM)	A(LP1)	A(LP2)	A(max)
Bus 1: Q	-2.9950	-0.2202	-0.2395	-0.5505	-0.2219	2.9950
Bus 2: Q	-2.9950	0.5350	0.4802	0.5445	0.5343	2.9950
Bus 3: Q	-2.9950	0.3534	0.3273	0.1951	0.3547	2.9950
Bus 4: Q	-2.9950	0.2539	0.4029	0.8000	0.2877	2.9950
Bus 5: Q	-2.9950	0.4475	0.4063	0.3352	0.4299	2.9950
Bus 6: Q	0.9450	0.5015	0.4474	0.3862	0.4637	1.1040
Bus 7: V	0.9450	1.0285	1.0320	1.0500	1.0288	1.1040
Bus 8: V	0.9450	1.0218	1.0261	1.0480	1.0222	1.1040
Bus 9: V	0.9450	1.0150	1.0204	1.0489	1.0147	1.1040
Bus 10: V	0.9450	1.0048	1.0080	1.0282	1.0046	1.1040
Bus 11: V	0.9450	0.9960	1.0039	1.0290	1.0007	1.1040
Bus 12: V	0.9450	0.9688	0.9825	1.0089	0.9789	1.1040
Bus 13: V	0.9450	1.0054	1.0125	1.0352	1.0104	1.1040
Bus 14: V	0.9450	0.9864	0.9963	1.0197	0.9939	1.1040
Bus 15: V	0.9450	0.9787	0.9910	1.0144	0.9884	1.1040
Bus 16: V	0.9450	0.9824	0.9923	1.0165	0.9895	1.1040
Bus 17: V	0.9450	0.9672	0.9798	1.0055	0.9765	1.1040
Bus 18: V	0.9450	0.9642	0.9772	1.0019	0.9742	1.1040
Bus 19: V	0.9450	0.9589	0.9723	0.9979	0.9691	1.1040
Bus 20: V	0.9450	0.9619	0.9755	1.0016	0.9722	1.1040
Bus 21: V	0.9450	0.9552	0.9725	0.9981	0.9689	1.1040
Bus 22: V	0.9450	0.9558	0.9744	0.9995	0.9707	1.1040
Bus 23: V	0.9450	0.9593	0.9807	1.0028	0.9776	1.1040
Bus 24: V	0.9450	0.9425	0.9759	0.9962	0.9972	1.1040
Bus 25: V	0.9450	0.9391	1.0138	1.0188	1.0093	1.1040
Bus 26: V	0.9450	0.9191	0.9946	0.9996	0.9901	1.1040
Bus 27: V	0.9450	0.9464	1.0466	1.0420	1.0415	1.1040
Bus 28: V	0.9450	1.0089	1.0110	1.0461	1.0043	1.1040
Bus 29: V	0.9450	0.8957	0.9959	0.9913	0.9909	1.1040
Bus 30: V	0.9450	0.8544	0.9546	0.9500	0.9496	1.1040

Table 6.17: Dependent Variables (Case VVC7)

Figures 6.19 and 6.20 present a subset of information in Table 6.17 graphically. The horizontal lines from bus number 7 to 30 give the minimum and maximum values of load bus voltages. Buses 1 to 6 are the generator buses. These figures allow an examination of how the dependent variables are adjusted from the initial state to the final state for different techniques.

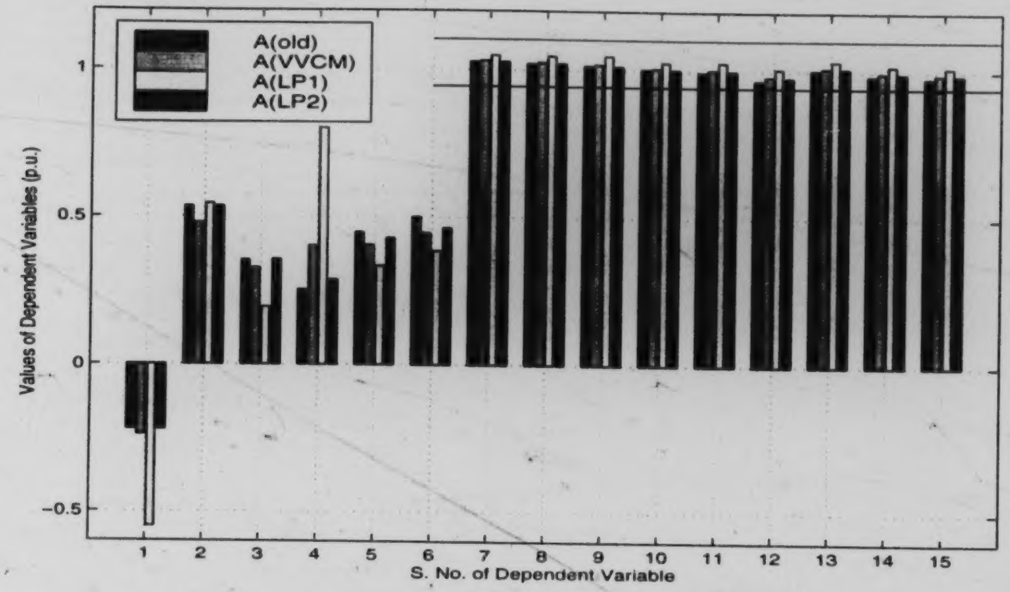


Figure 6.19: Dependent Variables, Buses 1-15 (Case VVC7)

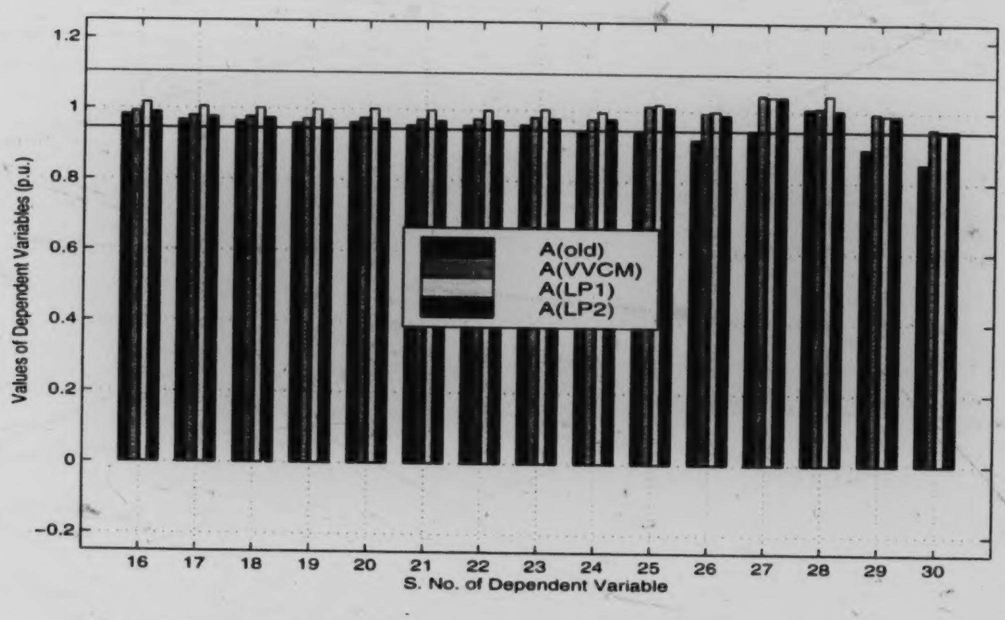


Figure 6.20: Dependent Variables, Buses 16-30 (Case VVC7)

6.1.1 Summary on VVCM

In this section, the performance of the VVCM algorithm, developed in this thesis was evaluated for various conditions of under- and over-voltages for a six bus system, IEEE 14 and IEEE 30 bus systems. A comparison was also made with two linear programming formulations (methods LP1 and LP2). The cases were:

1. 6 bus system for full-load.
2. 6 bus system at half of full-load.
3. 6 bus system at quarter of full-load.
4. IEEE 14 bus system loaded at 3.2 times full load
5. IEEE 30 bus system loaded at full load.
6. IEEE 30 bus system loaded at 2.1 times full load.
7. IEEE 30 bus system with a line contingency.

The main conclusions are:

1. The VVCM algorithm, developed in this thesis consistently performs better than other methods. The criteria was NSTEPS(R), that is the sum of absolute number of step changes for the controllers. This criteria has been extensively used in literature. However, a disadvantage of this method is that it treats generator voltage control as discrete. In this context, an additional criteria is that a minimum number of controllers should be changed. The VVCM performs the best using this criteria, as this method exhausts the most effective controller first.
2. Although the LP(2) method gives comparable result with the VVCM, it is most time consuming of all the methods, and may not be suitable for a fast action desired during power system problems. However, it (LP2) fits in the developed strategy, in that it can be used to optimize the response once situations demanding immediate attention have been taken care of by sub-optimal measures.

6.2 Load-Shedding

Background

In this section the performance of the load-shedding algorithms described in the previous chapter are evaluated.

Three load-shedding methods were described in the previous chapter. They are:

1. **Load-shedding Method (LSM-SC)** based on inverse of Sensitivity Coefficients (SC).
2. **Load-shedding Method (LSM-LP)** based on Linear Programming (LP).
3. **Load-shedding Method (LSM-DI)** based on computing Δ Current. (DI) injections required to ensure desired voltage profile over the system.

Abbreviations

Following abbreviations are used in the following discussion:

- X(min) = Minimum value of the control variables
- XLS(min) = Recommended value of load-shedding for minimum voltage at all the load buses
- XLS(nominal) = Recommended value of load-shedding for nominal voltage (typically 1 p.u.) at all the load buses
- XLS(normal) = Recommended value of load-shedding for normal voltage (within the specified range) at all the load buses
- XLS(max) = Recommended value of load-shedding for maximum voltage at all the load buses
- XLS(LP1) = Recommended value of load-shedding by the linear programming method with the objective function to be minimized given by:

$$f(x) = nq_1 + \ldots nq_m \tag{6.7}$$

where $(nq_1 \ldots nq_m)$ stand for step changes for m number of reactive loads

- XLS(LP2) = Recommended value of load-shedding by the linear programming method with the objective function to be minimized given by:

$$f(x) = |nq_1| + ...|nq_m| \tag{6.8}$$

where $|nq_1|...|nq_m|$ stand for absolute number of step changes for m number of reactive loads

- XLS(SC) = Recommended value of load-shedding based on inverse of sensitivity coefficient method
- XLS(SC-H) = Recommended value of load-shedding based on inverse of sensitivity coefficient method and supplemented by a heuristic
- X(max) = Maximum value of the control variables
- A(min) = Minimum values of the dependent variables
- ALS(min) = Predicted values of the dependent variables for XLS(min) load-shed
- ALS(nominal) = Predicted values of the dependent variables for XLS(nominal) load-shed
- ALS(normal) = Predicted values of the dependent variables for XLS(normal) load-shed
- ALS(max) = Predicted values of the dependent variables for XLS(max) load-shed
- ALS(LP1) = Predicted values of the dependent variables for XLS(LP1) load-shed
- ALS(LP2) = Predicted values of the dependent variables for XLS(LP1) load-shed
- ALS(SC) = Predicted values of the dependent variables for XLS(SC) load-shed
- ALS(SC-H) = Predicted values of the dependent variables for XLS(SC-H) load-shed
- A(max) = Maximum values of the dependent variables
- Sum_LS = Sum of reactive power load-shed

1. **Case LS1:** This case is for a 6 bus system. The system is loaded at 1.5 times the full-load level; settings of transformer taps and generator voltages are set at their maximum settings. The system has under-voltage violations at bus numbers 4, 5

and 6. The voltages are 0.8648, 0.8896 and 0.8279 p.u.. The desired minimum and maximum voltages at all the load buses are 0.895 p.u. and 1.045 p.u.. Since there are no resources left to solve voltage/VAr violations for this case, the system recommends load-shedding. Table 6.18 presents the following suggestions for load-shedding, using (LSM-SC) (load-shedding method based on inverse of sensitivity coefficients).

Type	X(min)	XLS(min)	XLS(normal)	XLS(nominal)	XLS(max)	X(max)
Bus 3: Q	0.0	-0.3939	-0.1912	-0.3911	-0.3884	0.195
Bus 4: Q	0.0	0.3750	0.1869	0.5858	0.7966	0.000
Bus 5: Q	0.0	-0.1612	-0.1612	-0.0895	-0.0179	0.270
Bus 6: Q	0.0	0.3462	0.3462	0.4785	0.6107	0.075

Table 6.18: Recommended Values for Load Shedding (Case LS1)

Entries in Table 6.18 are explained as follows:

- *Type* identifies the bus number and the variable under consideration. Thus *Bus 3: Q.* identifies the reactive power at bus number 3.
- *X(min)* is the minimum value of the control variables.
- *XLS(min)* is the recommended value of load-shedding to obtain $V_l(min)$ profile in the system.
- *XLS(normal)* is the recommended value of load-shedding to obtain $V_l(normal)$ profile in the system.
- *XLS(nominal)* is the recommended value of load-shedding to obtain $V_l(nominal)$ profile in the system.
- *XLS(max)* is the recommended value of load-shedding to obtain $V_l(max)$ profile in the system.
- *X(max)* is the maximum value of the control variables.

Figure 6.21 presents the information in Table 6.18 graphically.

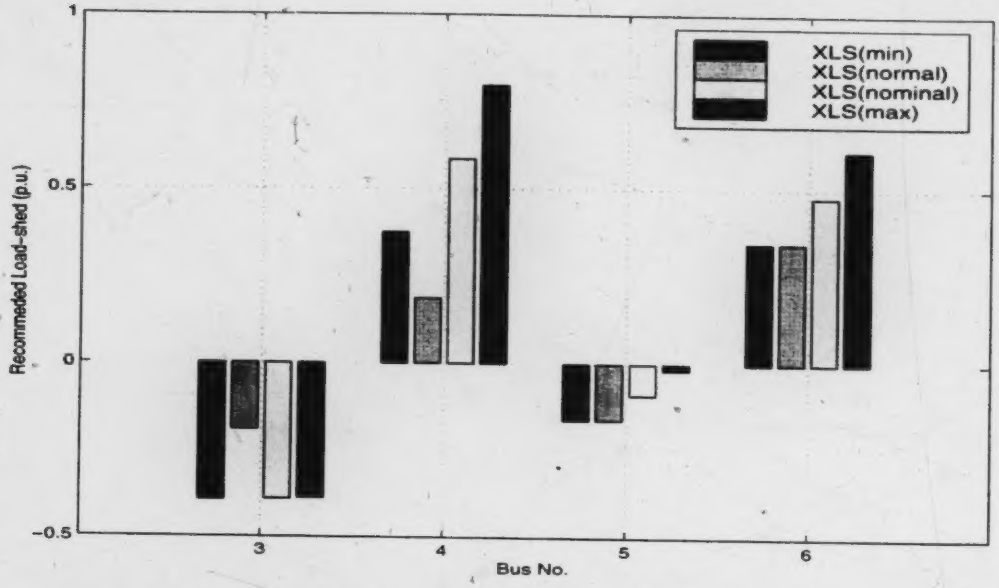


Figure 6.21: Recommended Values for Load Shedding (Case LS1)

Table 6.19 give the predicted values for the dependent variables for load-shedding recommendations, by different techniques presented in Table 6.18. A nominal voltage profile of 0.97 p.u. is chosen.

Type	A(min)	ALS(min)	ALS(normal)	ALS(nominal)	ALS(max)	A(max)
Bus 1: Q	-0.1955	0.9294	0.9294	0.5565	0.1836	1.0004
Bus 2: Q	-0.1955	0.4844	0.4557	0.2934	0.1025	1.0004
Bus 3: V	0.8950	0.8950	0.9268	0.9700	1.0450	1.0450
Bus 4: V	0.8950	0.8950	0.8950	0.9700	1.0450	1.0450
Bus 5: V	0.8950	0.8950	0.8950	0.9700	1.0450	1.0450
Bus 6: V	0.8950	0.8950	0.8950	0.9700	1.0450	1.0450

Table 6.19: Predicted Values for Load Shedding (Case LS1)

Entries in Table 6.19 are explained as follows:

- *Type* identifies the bus number and the variable under consideration. Thus *Bus 3: V*, identifies the voltage at bus number 3.
- *A(min)* is the minimum value of the dependent variables.

- *ALS(min)* is the predicted value of the dependent variables for XLS(min) load-shed to obtain $V_l(min)$ profile in the system.
- *ALS(normal)* is the predicted value of the dependent variables for XLS(normal) load-shed to obtain $V_l(normal)$ profile in the system.
- *ALS(nominal)* is the predicted value of the dependent variables for XLS(nominal) load-shed to obtain $V_l(nominal)$ profile in the system.
- *ALS(max)* is the predicted value of the dependent variables for XLS(max) load-shed to obtain $V_l(max)$ profile in the system.
- *A(max)* is the maximum value of the dependent variables.

Figure 6.22 presents the information in Table 6.19 graphically.

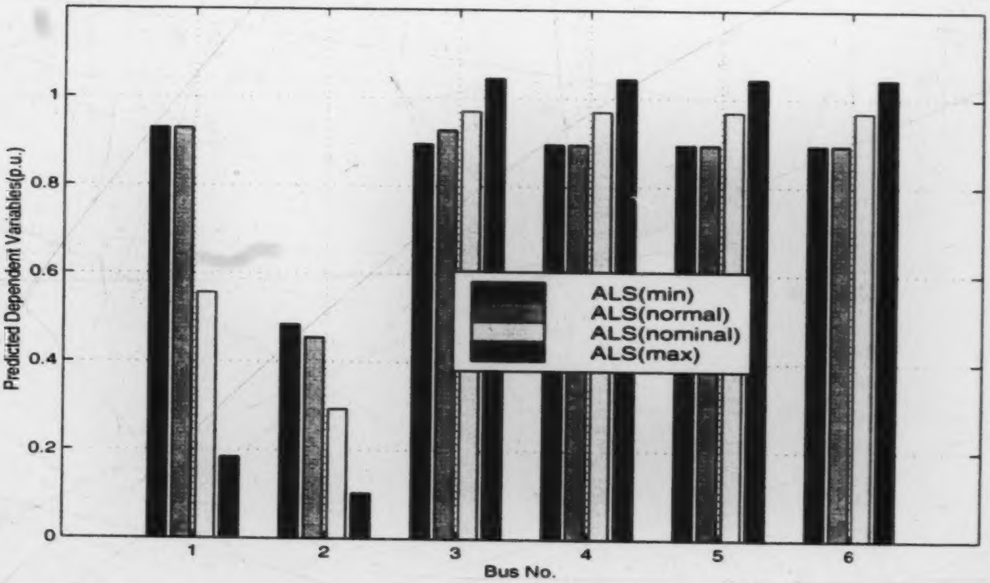


Figure 6.22: Predicted Values of Dependent Variables for Load Shedding (Case LS1)

Table 6.20 shows how the use of *heuristics* can help in minimizing the load-shedding. First of all it is assumed that a *normal* voltage profile is desired. XLS(normal) in Table 6.20, stands for reactive power loads which have to be shed to get a normal voltage profile over the entire system. It is computed by using the inverse of sensitivity coefficient method to get $V_l(normal)$ voltage profile. The column XLS(SC-H) stands for load which has to be shed, whose value is obtained by applying heuristic on

XLS(normal) values. In Table 6.18, the required XLS(SC) values for buses 3, 4, 5 and 6 are -0.1912, 0.1869, -0.1612 and 0.3462 p.u. respectively. From data of the 6 bus system (given in Appendix 'B'), one notes that buses 3, 4, 5 and 6 are connected by transmission lines and transformers. Also, buses 4, 5 and 6 are low voltage buses which are connected to each other forming a *low voltage area* [Vadari91]. Hence change in the reactive power requirements at buses 4, 5 and 6 may be combined. The combined value is 0.3719 p.u.. which is more than the installed value at these buses. These errors can be attributed to the linearization of sensitivities and large deviations from the operating point. A more accurate value may be obtained by using the inverse of reduced Jacobian matrix, J_R defined by equation 5.4.

Using J_R, the suggested values for XLS(normal) for buses 3, 4, 5 and 6 are: -0.1901, 0.1681, -0.1641 and 0.3157 p.u.. Now applying the heuristic of buses being connected together, the final value of reactive power load which must be shed is 0.3197 p.u.. This combined load-shed is distributed in proportion to existing reactive power loads at buses 4, 5 and 6.

The above heuristic closely follows from the concept of *low voltage area identification* suggested by [Vadari91].

Type	XLS(normal)	XLS(SC-H)	XLS(LP1)	XLS(LP2)	X(max)
Bus 3: Q	-0.1912	0.00	0.0103	0.0095	0.195
Bus 4: Q	0.1869	0.00	0.00	0.00	0.00
Bus 5: Q	-0.1612	0.2488	0.2335	0.2377	0.270
Bus 6: Q	0.3462	0.0691	0.0750	0.0750	0.075
Sum_LS		0.3179	0.3188	0.3222	

Table 6.20: Chosen Values for Load-Shedding (Case LS1)

Load-shedding methods based on sensitivity coefficient (supplemented by a heuristic) (indicated by XLS(SC-H)) presents the best solution, based on the criteria that least amount of load should be shed. However, the heuristic plays a crucial role. Load-shedding method (LP1) gives comparable solution and is much easier to apply.

Figure 6.23 presents the information in Table 6.20 graphically.

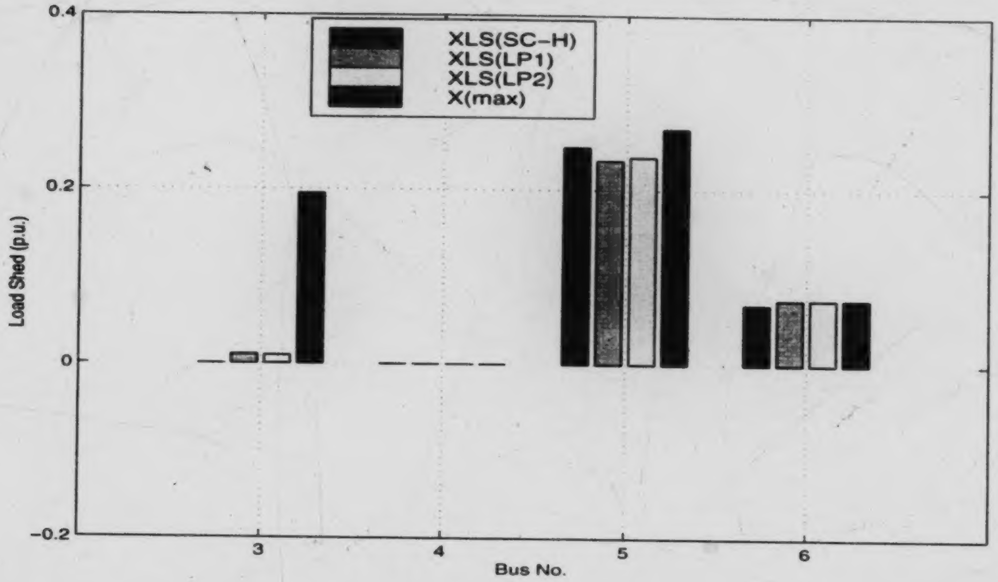


Figure 6.23: Chosen Values for Load-Shedding (Case LS1)

Table 6.21 compares values of the dependent variables predicted by the LSM-SC(H) (load-shedding method based on inverse of sensitivity coefficients and supplemented by a heuristic), and the linear programming methods 1 and 2 (LSM-LP1 and LSM-LP2) with load flows. Predicted values due to the application of LSM-SC(H), LSM-LP1 and LSM-LP2 are shown under ALS(SC-H), ALS(LP1) and ALS(LP2) respectively.

Type	ALS(LF-SCH)	ALS(SC-H)	ALS(LF-LP1)	ALS(LP1)	ALS(LF-LP2)	ALS(LP2)
Bus 1: Q	0.8802	0.9343	0.8730	0.9294	0.8711	0.9274
Bus 2: Q	0.2162	0.2454	0.2186	0.2495	0.2160	0.2467
Bus 3: V	0.9625	0.9715	0.9664	0.9584	0.9664	0.9585
Bus 4: V	0.9018	0.9113	0.9040	0.8950	0.9041	0.8952
Bus 5: V	1.0111	1.0340	1.0076	1.0031	1.0091	1.0048
Bus 6: V	0.9045	0.9212	0.9049	0.8950	0.9055	0.8957

Table 6.21: Predicted vs Load Flow Values of Dependent Variables (Case LS1)

Figure 6.24 presents an error bar graph showing the error between the predicted values of dependent variables for different techniques and load flows. This information has been derived from Table 6.21.

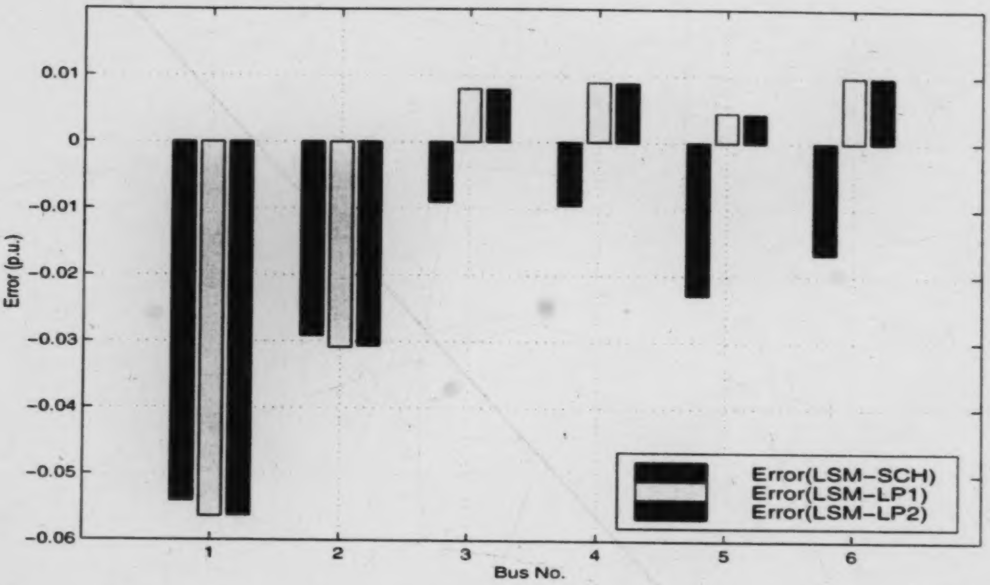


Figure 6.24: Error between Predicted and Load Flow Values of Dependent Variables (Case LS1)

Errors for different techniques are less than 0.06 p.u..

2. **Case LS2:** This case is for IEEE 14 bus system. The system is loaded at full-load level. There are under-voltage violations at bus numbers 9, 10 and 14 where the voltages are 0.9920, 0.9916 and 0.9798 p.u. respectively. The desired minimum and maximum voltages at all the load buses are 1.0000 and 1.0500 p.u. respectively. Assuming that there are no resources left to solve voltage/VAr violation for the case, the system recommends load-shedding. Table 6.22 presents following suggestions for load-shedding using (LSM-SC), a method based on inverse of sensitivity coefficients.

Type	XLS(min)	XLS(normal)	XLS(nominal)	XLS(max)
Bus 6: Q	0.1905	-0.0144	0.4538	0.7171
Bus 7: Q	-0.3074	0.0000	0.0357	0.3787
Bus 8: Q	-0.2398	-0.0735	-0.0966	0.0465
Bus 9: Q	0.1189	0.0386	0.1077	0.0965
Bus 10: Q	0.0866	0.0436	0.0865	0.0864
Bus 11: Q	-0.1257	-0.0404	-0.0113	0.1031
Bus 12: Q	-0.0822	0.0000	0.0041	0.0904
Bus 13: Q	-0.1195	-0.0527	0.0490	0.2175
Bus 14: Q	0.1157	0.0912	0.1157	0.1157

Table 6.22: Recommended Values for Load Shedding (Case LS2)

Figure 6.25 presents the information in Table 6.22 graphically.

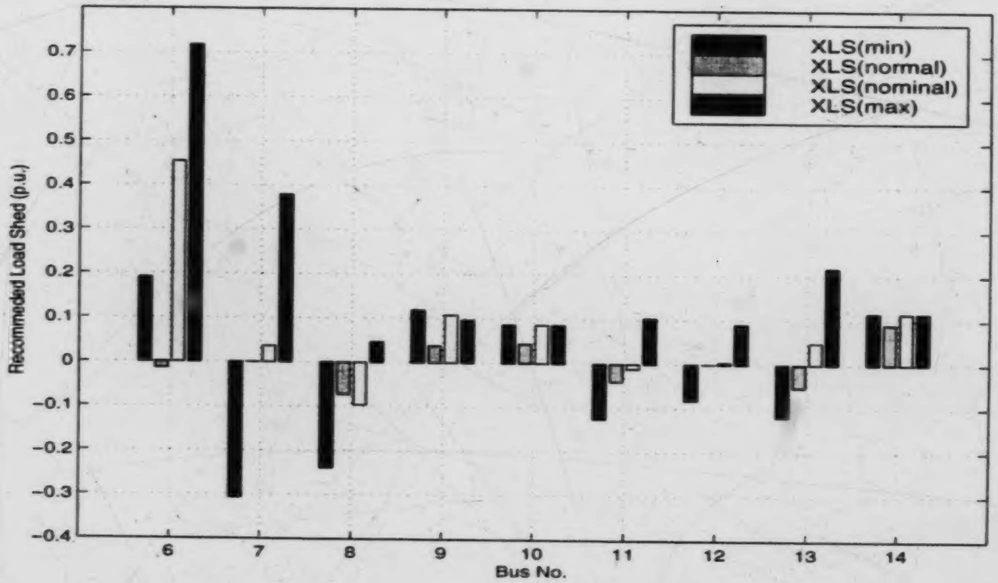


Figure 6.25: Recommended Values for Load Shedding (Case LS2)

In this case, a heuristic is applied which is different than the one used in the previous case (Case LS1). This heuristic may be explained as follows: As indicated in Chapter 5, the values of $\Delta Q_l(normal)$ and $\Delta Q_l(max)$ provide the safe range of ΔQ_l values which may be shed. Thus, for a bus, any value of reactive power load in the above mentioned range may be shed. In order to minimize the load shed and the number of changes, a zero value (that is, no load-shedding) is the best value. Also since buses

13 and 14 are connected, the suggested reactive power load sheds at these buses are combined to get a value of 0.0385 p.u. at bus 14. Applying these heuristics, Table 6.23 presents the following chosen values for load-shedding using the sensitivity coefficient method for load-shedding (LSM-SC) and supplemented by a heuristic. This value is shown under XLS(SC-H). Th's Table also presents computed values of load-shedding using linear programming methods 1 and 2 (LSM-LP1 and LSM-LP2). The chosen values for load-shedding using methods LSM-LP1 and LP2 are shown under XLS(LP1) and XLS(LP2) respectively

Type	XLS(SC-H)	XLS(LP1)	XLS(LP2)	X(max)
Bus 6: Q	0.00	0.00	0.00	0.039
Bus 7: Q	0.00	0.00	0.00	0.018
Bus 8: Q	0.00	0.00	0.00	0.000
Bus 9: Q	0.0386	0.09456	0.09660	0.166
Bus 10: Q	0.0436	0.00	0.00	0.058
Bus 11: Q	0.00	0.00	0.00	0.018
Bus 12: Q	0.00	0.00	0.00	0.016
Bus 13: Q	0.00	0.00	0.00	0.058
Bus 14: Q	0.0385	0.0560	0.0560	0.056
Sum_LS	0.1207	0.1506	0.1526	

Table 6.23: Chosen Values for Load-Shedding (Case LS2)

One notes that LSM-SC, a method developed in this thesis provides the best solution based on the criteria of minimizing the amount of load to be shed. But heuristics are needed. Method LSM-LP1 is easiest to apply.

Figure 6.26 presents the information in Table 6.23 graphically.

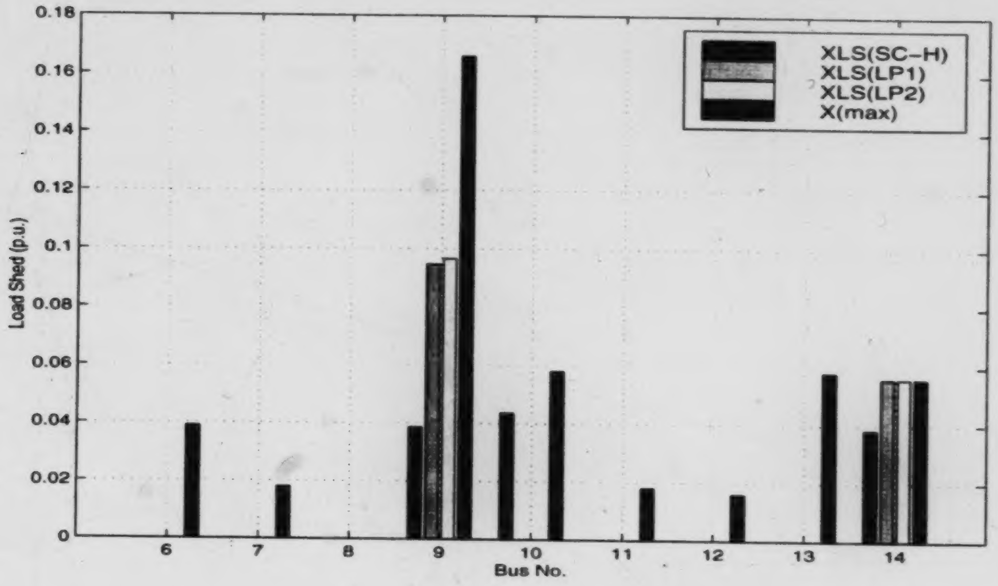


Figure 6.26: Chosen Values for Load-Shedding (Case LS2)

Table 6.24 compares the predicted values of the dependent variables by the application of load-shedding method-SC (indicated under ALS(SC-H)), linear programming based load-shedding formulations (indicated under ALS(LP1) and ALS(LP2)) with load flows.

Type	ALS(LF-SCH)	ALS(SC-H)	ALS(LF-LP1)	ALS(LP1)	ALS(LF-LP2)	ALS(LP2)
Bus 1: Q	0.3080	0.3081	0.3064	0.3070	0.3063	0.3069
Bus 2: Q	-0.1411	-0.1415	-0.1468	-0.1461	-0.1472	-0.1464
Bus 3: Q	0.4161	0.4159	0.4127	0.4130	0.4125	0.4128
Bus 4: Q	0.3710	0.3699	0.3573	0.3593	0.3564	0.3584
Bus 5: Q	0.1767	0.1749	0.1641	0.1667	0.1634	0.1660
Bus 6: V	1.0028	1.0029	1.0035	1.0033	1.0036	1.0033
Bus 7: V	1.0102	1.0145	1.0106	1.0106	1.0106	1.0106
Bus 8: V	1.0142	1.0035	1.0163	1.0159	1.0164	1.0160
Bus 9: V	1.0029	1.0044	1.0071	1.0063	1.0073	1.0065
Bus 10: V	1.0037	1.0044	1.0041	1.0035	1.0043	1.0037
Bus 11: V	1.0153	1.0156	1.0155	1.0151	1.0156	1.0152
Bus 12: V	1.0184	1.0185	1.0192	1.0190	1.0192	1.0190
Bus 13: V	1.0129	1.0132	1.0145	1.0141	1.0145	1.0141
Bus 14: V	0.9934	0.9944	1.0000	0.9986	1.0001	0.9987

Table 6.24: Predicted vs Load Flow Values of Dependent Variables (Case LS2)

Figure 6.27 presents an error bar graph showing the error between the predicted values of the dependent variables for different techniques and load flows. This information has been derived from Table 6.24.

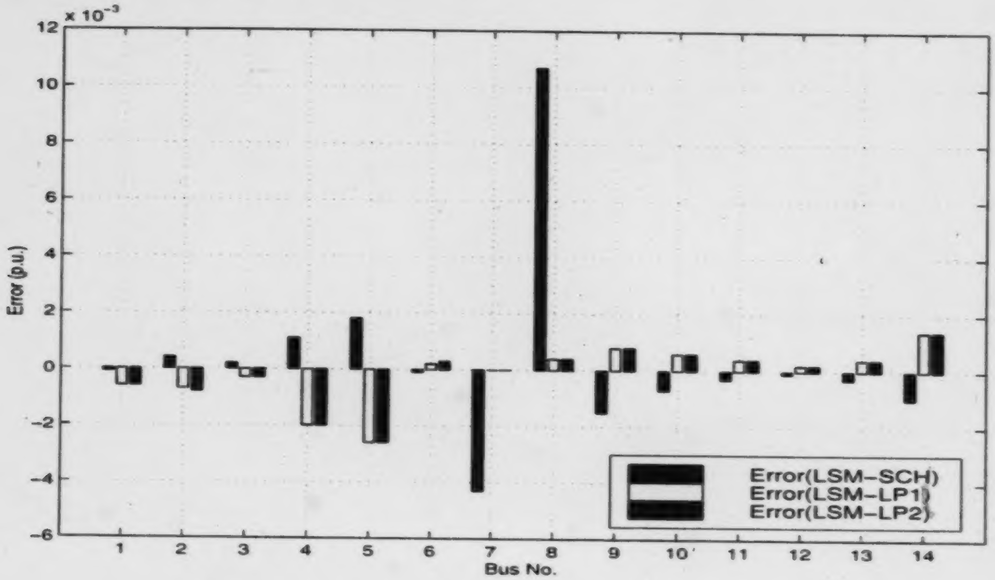


Figure 6.27: Error between Predicted and Load Flow Values of Dependent Variables (Case LS2)

All errors are less than 0.012 p.u..

3. **Case LS3:** This case is for IEEE 14 bus system. The system is loaded at 3.2 times[3] the full-load level. This case is the same as VVC6 (described in section 6.1), with a modification that in this simulated case, all transformer taps and all generator voltages are at their upper limits. The system has under-voltage violations at bus numbers 6, 7, and 14 where the voltages are 0.9387, 0.9492 and 0.9263 p.u. respectively. The desired minimum and maximum voltages at all the load buses are 0.9500 and 1.0450 p.u. respectively. Since, all the resources to solve voltage/VAr violation for this case have been exhausted, the system recommends load-shedding.

---

[3] Loading of 3.2 times the full-load, does not reflect the actual power system operational scenario. At these load levels, relays would trip in order to save the transmission lines and transformers. In these studies, it is just a means to artificially stress the system to induce voltage/VAr violations. This is an alternative method for inducing voltage/VAr violations in place of simulating line and generator contingencies. In this author's opinion, a possible advantage of using this way of stressing the system is that sensitivities need not be recalculated as the topology of the connected system does not change. On the other hand, with a line and/or generator contingency, the system topology changes, which necessitates recalculation of sensitivities.

Table 6.25 presents following chosen values for load-shedding using linear programming methods 1 and 2.

Type	XLS(LP1)	XLS(LP2)	X(max)
Bus 6: Q	0.1365	0.1365	0.1365
Bus 7: Q	0.0630	0.0630	0.0630
Bus 8: Q	0.00	0.00	0.00
Bus 9: Q	0.1328	0.1022	0.5810
Bus 10: Q	0.00	0.00	0.2030
Bus 11: Q	0.00	0.00	0.0630
Bus 12: Q	0.00	0.00	0.056
Bus 13: Q	0.00	0.00	0.2030
Bus 14: Q	0.0445	0.0541	0.1960
Sum_LS	0.3768	0.3558	

Table 6.25: Chosen Values for Load Shedding (Case LS3)

Method LSM-LP2 provides slightly better result than LSM-LP1, but is more time consuming.

Figure 6.28 presents the information in Table 6.25 graphically.

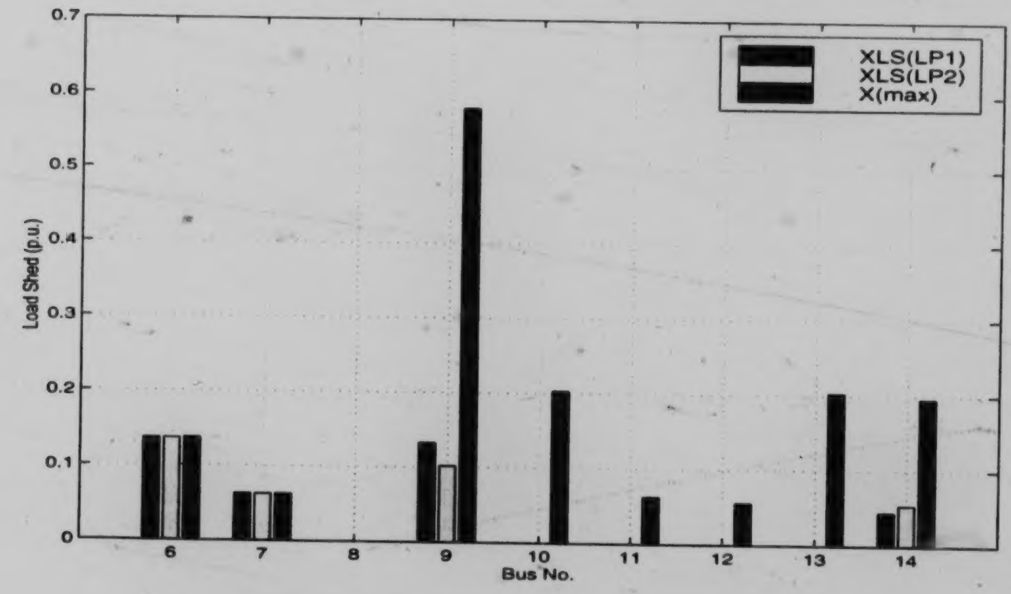


Figure 6.28: Chosen Values for Load Shedding (Case LS3)

Table 6.26 compares the predicted values of the dependent variables by load-shedding methods based on linear programming methods 1 (LSM-LP1) and 2 (LSM-LP2) respectively with load flows.

Type	ALS(LF-LP1)	ALS(LP1)	ALS(LF-LP2)	ALS(LP2)
Bus 1: Q	0.0034	0.0287	0.0102	0.0301
Bus 2: Q	5.6286	5.7175	5.6489	5.7219
Bus 3: Q	3.3674	3.4058	3.3778	3.4085
Bus 4: Q	2.2319	2.2980	2.2740	2.3052
Bus 5: Q	0.3689	0.3932	0.3949	0.4027
Bus 6: V	0.9527	0.9500	0.9513	0.9495
Bus 7: V	0.9614	0.9582	0.9602	0.9579
Bus 8: V	1.0409	1.0370	1.0368	1.0355
Bus 9: V	1.0199	1.0138	1.0121	1.0109
Bus 10: V	1.0104	1.0014	1.0004	0.9990
Bus 11: V	1.0422	1.0375	1.0370	1.0363
Bus 12: V	1.0435	1.0428	1.0431	1.0428
Bus 13: V	1.0237	1.0225	1.0229	1.0226
Bus 14: V	0.9551	0.9500	0.9516	0.9501

Table 6.26: Predicted vs Load Flow Values of Dependent Variables (Case LS3)

Figure 6.29 presents an error bar graph showing the error between the predicted values of the dependent variables for different techniques and load flows. This information has been derived from Table 6.26.

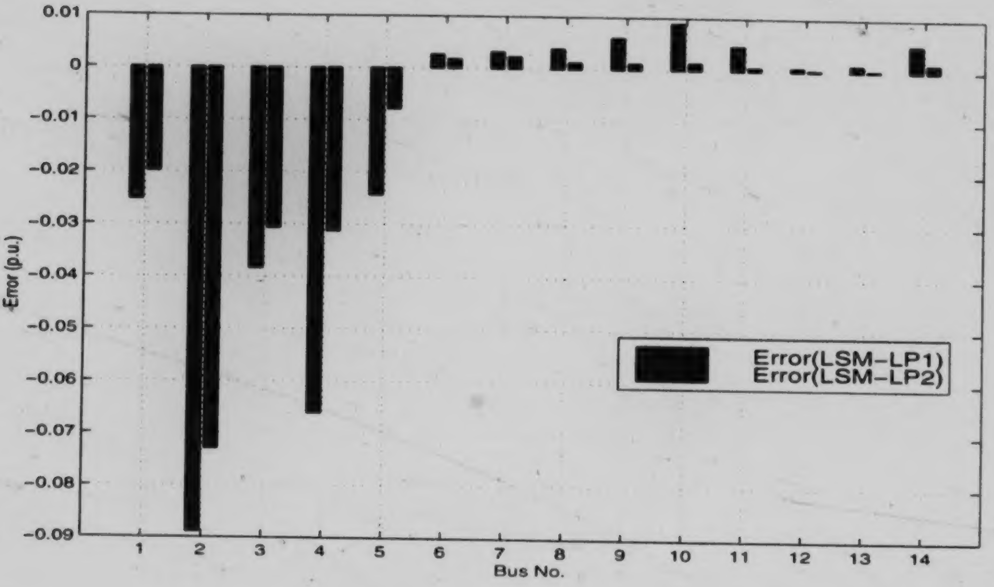


Figure 6.29: Error between Predicted and Load Flow Values of Dependent Variables (Case LS3)

All errors are less than 0.09 p.u..

4. **Case LS4:** This case is for a two bus system. This example demonstrates the working of load-shedding method based on Δ current (DI) injection method (LSM-DI). Bus 1 is the generator bus and bus 2 is the load bus.

This system has real and reactive power loads of 6 p.u. each. The voltage at bus 2 is 0.9271 p.u.. To achieve a voltage of 0.945 p.u. at the load bus, the system recommends that compensating current injections of (-0.0736 + j1.7782) and (0.1910 - j1.9101) be made at buses 1 and 2 respectively. This translates to a compensation impedance of (-0.0774 - j0.4861) p.u. at bus 2. At bus 1 the compensating current injection lead to a change in the internal voltage behind the reactance of the generator from (1.0791 - j0.0461) p.u. to (1.0613 - j0.0468) p.u.. Generator reactance of 0.01 p.u. is assumed.

Thus the new power recommended at bus 2 is (5.9484 + j4.4422) p.u.. The load flow was run. On checking, the voltage at bus 2 is 0.945 p.u..

Discussion: This method gives accurate results. However, this method requires that the desired voltages at the load buses be supplied along with the angle information. This can be problematic. On the positive side, this method can take into account the presence of the phase shifters and the relationship between power and voltages. Also, as shown above, the delta current injections can show the effect of load-shedding on generators. More research is required to develop this methodology.

6.2.1 Summary on Load-shedding

In the previous section, the performance of load-shedding algorithms was evaluated for various conditions for a six bus system and IEEE 14 bus systems. A comparison was also made with two linear programming formulations. The cases were:

1. 6 bus system loaded at full load level. Use of a heuristic is illustrated.
2. IEEE 14 bus system loaded at full load level. Use of a heuristic is illustrated.
3. IEEE 14 bus system loaded at 3.2 times the full load level.
4. A 2 bus system.

The main conclusions are:

1. The load-shedding algorithms (LSM-SC), (LSM-LP1), and (LSM-LP2) developed in this thesis, perform well, as indicated by comparison of results with load flows.
2. The load-shedding algorithm based on linear programming method 1 (LSM-LP1) is the best based on the criteria of minimizing the load-shed and ease of application. Although linear programming methods tend to be computationally intensive, it may be mentioned that linear programming formulation for load-shedding is a subset of the voltage and VAr control problem. Linear programming performance depends upon the numbers of variables. For voltage and VAr control problem the number of variables involved are equal to number of transformers, twice the number of generators, and twice the number of load buses. In this study, the number of variables are equal to number of generators, and twice the number of load buses.
3. The load-shedding method (LSM-SC) performance depends upon the use of heuristics.
4. The load-shedding algorithm (LSM-DI) needs further research.
5. The developed load-shedding algorithms in this research are able to predict their action on the rest of the system, which is an improvement over some existing methods.

6.3 Neural-Network Based Power Flows

Background

A neural network architecture called **ADA**ptive **LI**near **NE**uron (ADALINE)[Patterson96] has been used in this research to predict power flows over the transmission lines and transformers. The power flows are classified under five levels.

The input vector for predicting power flows (for example, between buses i and j) comprises of the following:

$$input = [(tap_1 ... tap_t); (V_i - V_j); (P_{gi} - P_{li}); (Q_{gi} - Q_{li}); (P_{gj} - P_{lj}); (Q_{gj} - Q_{lj})]; \quad (6.9)$$

where

tap_1 ... tap_t is the tap setting of t number of transformers

$(V_i - V_j)$ is the voltage difference between buses i and j

$(P_{gi} - P_{li})$ is the difference in real power generation and real power load at bus i

$(Q_{gi} - Q_{li})$ is the difference in reactive power generation and reactive power load at bus i

$(P_{gj} - P_{lj})$ is the difference in real power generation and real power load at bus j

$(Q_{gj} - Q_{lj})$ is the difference in reactive power generation and reactive power load at bus j.

The output is power flow level over a transmission line or a transformer connected between buses i and j.

The criterion for success for this study is that for all test cases, the maximum error allowed is one power flow level.

Abbreviations

In the following tables, following abbreviations are used:

- *From-To* indicates the interconnection between the transmission lines and transformers. Transformers are indicated by (T) along entries in the *From-To* column.
- PF(VVCM) = predicted power flow level after voltage/VAr correction by the VVCM.
- PF(LP1) = predicted power flow level after voltage/VAr correction by the linear programming method 1 (LP1).

- PF(LP2) = predicted power flow level after voltage/VAr correction by the linear programming method 2 (LP2).

- PF(LF) = actual power flow level obtained after running load flow.

1. **Case NN1:** This case is for 6 bus system. This case is the same as case VVC1 (in section 6.1) for voltage/VAr violations. After the settings of the controllers have been changed by different methods, for remedying voltage/VAr violations, new power flows emerge which are predicted by the neural network based system and the results compared with load flow.

Table 6.27 presents the results.

S. No.	From-To	PF(VVCM))	PF(LF-VVCM)	PF(LP1)	PF(LF-LP1)	PF(LP2)	PF(LF-LP2)
1.	1-6	2	2	3	3	2	2
2.	1-4	2	2	3	3	2	2
3.	4-6	2	2	1	1	2	2
4.	5-6(T)	2	1*	1	2	2	1*
5.	5-2	3	3	3	3	3	3
6.	2-3	2	2	2	2	2	2
7.	3-4(T)	2	2	3	3	2	2

Table 6.27: Predicted vs Power Flow Levels (Case NN1)

In this table “” besides a PF entry indicates that the load-flow result matched very closely with power flow levels predicted by the neural-network based system, even though a difference of one power flow level has been recorded.*

Figure 6.30 presents an error bar graph showing the error between the predicted values of power flow levels and load flows. This information has been derived from Table 6.27.

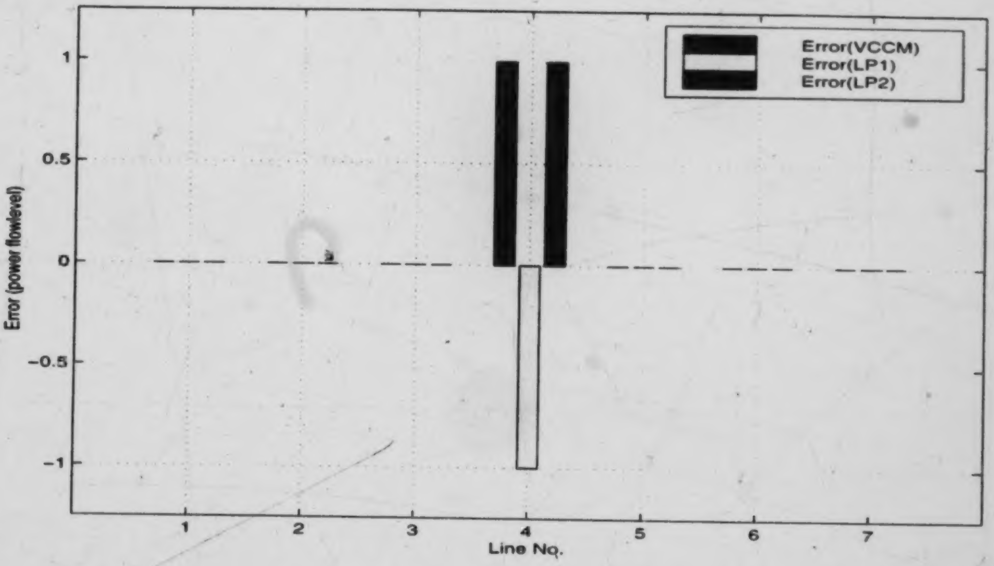


Figure 6.30: Error between Predicted Power Flow Levels and Load Flow Values (Case NN1)

The criterion that the maximum error allowed is one power flow level has been met.

2. **Case NN2:** This case is for IEEE 14 bus system. After the settings of the controllers have been changed by different methods, new power flows emerge which are predicted by the neural network based system and the results are compared with load flow.

Table 6.28 presents the results.

S. No.	From-To	PF(VVCM)	PF(LF-VVCM)	PF(LP1)	PF(LF-LP1)	PF(LP2)	PF(LF-LP2)
1.	1-2	4	4	4	4	4	4
2.	1-7	4	4	4	4	4	4
3.	2-3	4	4	4	4	4	4
4.	2-6	3	4	3	4	3	4
5.	2-7	3	4	3	4	3	4
6.	3-6	2	3*	3	3	3	3
7.	6-7(T)	3	4	4	4	3	4
8.	6-8(T)	3	4	3	4	3	4
9.	6-9(T)	3	3	3	3	3	3
10.	4-7	4	4	4	4	4	4
11.	4-11	3	3	3	3	3	3
12.	4-12	4	4	3	3	3	3
13.	4-13	3	3	3	4*	3	3
14.	8-5	3	3	3	3	3	3
15.	8-9	3	3	3	3	4	4
16.	9-10	2	2	2	2	2	2
17.	9-14	3	3	3	3	3	3
18.	10-11	3	3	3	3	3	3
19.	12-13	3	2	3	3	3	2*
20.	13-14	3	3	3	3	3	3

Table 6.28: Predicted vs Power Flow Levels (Case NN2)

In this table "" besides a PF entry indicates that the load-flow result matched very closely with power flow levels predicted by the neural-network based system, even though a difference of one power flow level has been recorded.*

Figure 6.31 presents an error bar graph showing the error between the predicted values of power flow levels and load flows. This information has been derived from Table 6.28.

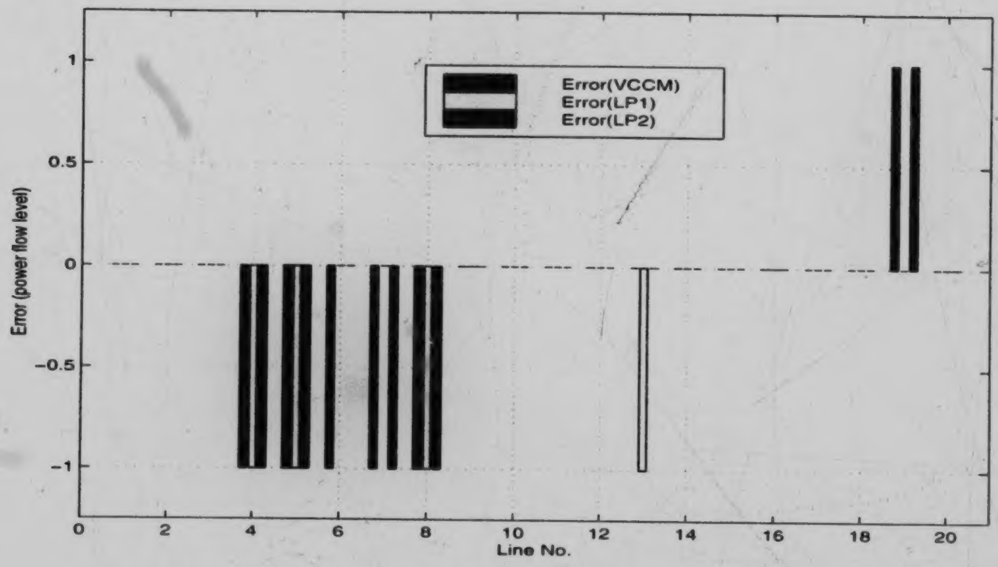


Figure 6.31: Error between Predicted Power Flow Levels and Load Flow Values (Case NN2)

The criterion that the maximum error allowed is one power flow level has been met.

3. **Case NN3:** This case is for IEEE 30 bus system. After the settings of controllers have been changed, new power flows emerge which are predicted by the neural network based system and the results compared with load flow. Two situations have been studied.

In the first case (Case A), the system was loaded upto 2.1 times the full load level, the settings of the controllers were changed randomly and the neural network based system was used to predict the new power flow levels. For many lines, the error was more than one power flow level which is not acceptable. This can be explained by the fact that while neural networks are good at interpolation, they are poor at extrapolation. The database for the IEEE 30 bus system was generated with maximum system loading at 2.1 times the full load level. Beyond that numerical instability results (as load flows do not converge). While generating the database (described in Chapter 5), the value of various variables (such as loads, generation, etc.) are multiplied by a random number (range 0 to 1, average value 0.5) to yield new values. This means that most of the entries in the database for system loading are at levels less than 2.1 times.

In the second case (Case B), the system was loaded around full load, and setting of the controllers changed randomly. In this case, the neural network using the database mentioned above gave satisfactory results, that is, the errors in the power flow levels are at the most one power flow level. Table 6.29 presents the results for the two cases mentioned above.

S. No.	From-To	PF(Case A)	PF(LF-A)	PF(Case B)	PF(LF-B)
1.	1-2	4	4	2	2
2.	1-7	4	4	2	3
3.	2-8	3	4	2	2
4.	7-8	3	5	3	3
5.	2-3	2	4	2	2
6.	2-9	3	5	3	3
7.	8-9	3	4	2	2
8.	3-10	3	2	2	1
9.	9-10	3	3	2	2
10.	9-4	3	3	2	1
11.	9-11(T)	2	4	3	2
12.	9-12(T)	2	4	2	2
13.	11-5	0	1	2	2
14.	11-12	3	4	2	2
15.	8-13(T)	3	4	3	2
16.	13-6	3	3	2	2
17.	13-14	3	4	2	2
18.	13-15	3	4	2	2
19.	13-16	2	2	2	2
20.	14-15	2	2	2	1
21.	16-17	1	1	2	2
22.	15-18	2	3	2	2
23.	18-19	2	2	2	1
24.	19-20	4	4	2	2
25.	12-20	4	4	2	2
26.	12-17	3	3	2	2
27.	12-21	3	4	2	2
28.	12-22	2	3	2	2
29.	21-22	3	4	2	1
30.	15-23	1	2	2	2
31.	22-24	2	4	2	2
32.	23-24	0	1	2	1
33.	24-25	3	2	2	1
34.	25-26	4	4	2	2
35.	25-27	2	3	2	2
36.	28-27(T)	3	5*	2	2
37.	27-29	3	4	2	2
38.	27-30	4	4	2	2
39.	29-30	3	4	2	2
40.	4-28	3	3	1	1
41.	9-28	3	5	3	2

Table 6.29: Predicted vs Power Flow Levels (Case NN3)

In this table "" besides a PF entry indicates that the load-flow result was close to power flow (PF) level of 4, even though a difference of two power flow levels has been recorded.*

Figures 6.32 and 6.33 present an error bar graph showing the error between the predicted values of power flow levels and load flows, for the two cases. This information has been derived from Table 6.29.

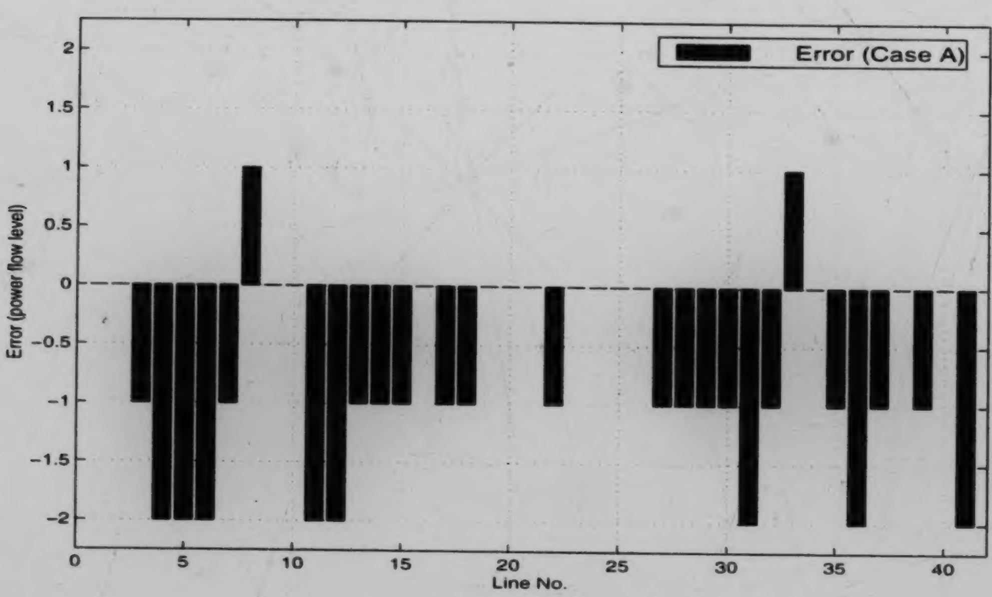


Figure 6.32: Error between Predicted Power Flow Levels and Load Flow Values (Case NN3)

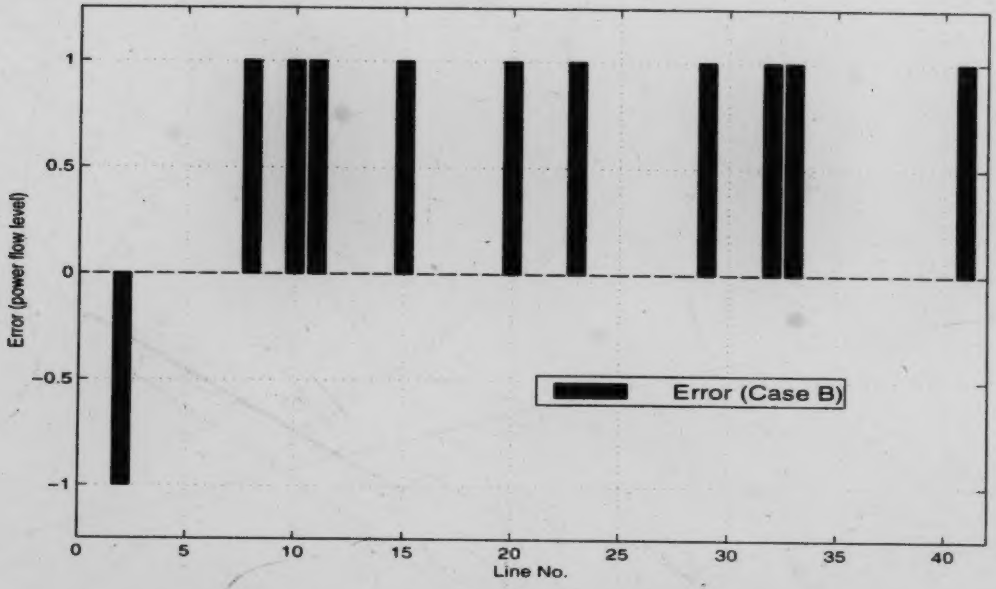


Figure 6.33: Error between Predicted Power Flow Levels and Load Flow Values (Case NN3)

6.3.1 Summary for Neural Network Based System

In this section, the performance of a neural network based power flow level predictor was evaluated for various conditions for a 6 bus, IEEE 14 and IEEE 30 bus systems. The cases were:

1. 6 bus system for full load.
2. IEEE 14 bus system for full load.
3. IEEE 30 bus system is tested for load at 2.1 times the full load level and at full load level.

The main findings are:

1. A properly trained neural network can give reliable and accurate results.
2. **In this thesis, a novel way of predicting power flow levels over the transmission lines and transformers based on the settings of the controllers has been presented. Used in conjunction with the VVCM and the load-shedding methods, it enhances the end product (the CVVC toolbox). While the VVCM and the load-shedding methods 1 and 2 are based on the de-coupled**

power system model, this system is able to predict apparent power flows over the transmission lines and transformers. This is achieved by using a simple neural network architecture and being trained in an off-line mode.

6.4 Data Mining Based Induction of Decision Trees

Background

The training set consists of tuples that comprises transformer tap ratio, generator voltage and real and reactive loads at the two load buses. All the input values are continuous. The induced decision tree classifies the power system into *Stable*, *IntStable* (intermediate stable) or *Unstable* mode depending upon the input pattern. The class *IntStable* was classified by examining the load bus voltages and the corresponding **Voltage Q** (reactive power) **Sensitivity (VQS)** values. For VQS values above a certain limit (with load flow converged) the power system operating mode was classified as *IntStable*.

In this section, examples are presented for inducing a decision tree and its use to classify the power system operating modes into *Stable*, *IntStable* (intermediate stable) or *UnStable* state. The program used is *c4.5* [Quinlan93], which is an extension of ID3 algorithm [Quinlan86] which is described in Chapter 3.

1. **Case DT1:** This case is for a 3 bus system. This example has been chosen because as the number of elements in a power system increases, the size of the tree grows very big. This makes comprehension of the generated decision tree difficult. This system has one transformer with tap, one generator bus, and two load buses. In this case, the power system operating mode is classified as *Stable* or *Unstable* depending on whether the load flow converged or not. The input comprises of transformer tap ratio, generator voltage and real and reactive loads at the two load buses. All the input values are continuous.

   The UNIX command *c4.5 -u -f bus3* invokes the *c4.5* program with the *-f* option giving the task name and the *-u* option indicating that the classifier is to be tested on unseen cases. A portion of the output of the decision-tree generator in this instance appears as Figure 6.34 (the complete tree is given in Appendix 'E'). An explanation of the results follows Figure 6.34.

```
------------------------------------------------------------
C4.5 decision tree generator
-----------------------------------------
    Options:
Trees evaluated on unseen cases
Read 2000 cases (8 attributes) from bus3.data

Decision Tree:
QL3 <= 1.9895 :
|   PL3 <= 3.0859 :
|   |   QL3 <= 1.3591 :
|   |   |   QL2 <= 3.1796 :
|   |   |   |   PL3 <= 1.8876 : Stable (331.0)
|   |   |   |   PL3 > 1.8876 :
|   |   |   |   |   VG1 <= 0.93 :
|   |   |   |   |   |   QL2 > 2.7538 : UnStable (5.0)
|   |   |   |   |   |   QL2 <= 2.7538 :
|   |   |   |   |   |   |   QL3 <= 0.6761 : Stable (17.0)
|   |   |   |   |   |   |   QL3 > 0.6761 :
|   |   |   |   |   |   |   |   QL2 <= 0.8336 : Stable (8.0)
|   |   |   |   |   |   |   |   QL2 > 0.8336 :
|   |   |   |   |   |   |   |   |   PL2 > 2.1731 : UnStable (7.0)
|   |   |   |   |   |   |   |   |   PL2 <= 2.1731 :
|   |   |   |   |   |   |   |   |   |   PL3 <= 2.6636 : Stable (2.0)
|   |   |   |   |   |   |   |   |   |   PL3 > 2.6636 : UnStable (3.0/1.0)
|   |   |   |   |   VG1 > 0.93 :
|   |   |   |   |   |   QL2 <= 2.7104 : Stable (141.0/1.0)
|   |   |   |   |   |   QL2 > 2.7104 :
|   |   |   |   |   |   |   VG1 > 1 : Stable (16.0)
|   |   |   |   |   |   |   VG1 <= 1 :
|   |   |   |   |   |   |   |   QL3 <= 0.433 : Stable (5.0)
|   |   |   |   |   |   |   |   QL3 > 0.433 : UnStable (8.0/1.0)
.....
..... <deleted>
```

```
.....

Simplified Decision Tree: <deleted>

Evaluation on training data (2000 items):

 Before Pruning          After Pruning
----------------   ---------------------------
Size      Errors   Size      Errors   Estimate

 195   24( 1.2%)    149   39( 1.9%)    ( 6.8%)   <<

Evaluation on test data (4165 items):

 Before Pruning          After Pruning
----------------   ---------------------------
Size      Errors   Size      Errors   Estimate

 195  174( 4.2%)    149  205( 4.9%)    ( 6.8%)   <<

  (a)  (b) <-classified as
 ---- ----
 2267  146 (a): class Stable
   59 1693 (b): class UnStable
-----------------------------------------------------------
```

Figure 6.34: Decision Tree (Case DT1)

Explanation: After a preamble recording the options used and the number of training cases read, the output gives the decision tree generated from the cases. The number in parentheses, following each leaf, indicates the number of training cases associated with each leaf and the number of them misclassified by a leaf.

After presenting the simplified tree, the output shows both the original and simplified trees performance on the training set from which they were constructed. The original tree of 195 nodes misclassifies 24 of the 2000 training cases. The simplified tree of 149 nodes misclassifies 39 training case. The program predicts that it will have an error rate of 6.8% on unseen cases. The "<<" mark focuses attention on the key results.

The decision tree classification model is now evaluated on the unseen test cases. The performance on unseen test cases is summarized as above.

The purpose of constructing classification models is not limited to the development of accurate predictors, although this is certainly a key concern. Another principal aim is that the model should be intelligible to human beings [Carbonell83, Madan97c][4]. In this thesis, the program *c4.5* is used to achieve this goal by re-expressing the classification model as ***production rules***, a format that appears to be more intelligible than trees [Quinlan93].

The programs use a simplified form of production rule $L \rightarrow R$ in which the left-hand side L is conjunction of attribute-based tests and the right-hand side R is a class. One of the classes is also designated as a *default*. To classify a case using a production rule model, the ordered list of rules is examined to find the first whose left-hand side is satisfied by the case. The predicted class is then the one nominated by the right-hand side of the rule. If no rule's left-hand side is satisfied, the case is predicted to belong to the default class.

The program *c4.5rules* examines the (original) decision tree produced by the *c4.5 program* and derives from it a set of production rules of the form above. The command *c4.5rules -u -f bus3* invokes the generation of production rules. *c4.5rules* produces the output, a portion of which is shown as Figure 6.35 below (the complete set of production rules is given in Appendix 'D'). An explanation of results follows:

---

[4] "Data Mining Applications for Power Systems," *VII th International Conference on Electronics, Communications and Computers, (CONIELECOMP'97)*, Puebla, Mexico, February 1997.

```
------------------------------------------------
C4.5 rule generator
--------------------------------
    Options: Rulesets evaluated on unseen cases

Read 2000 cases (8 attributes) from bus3.data
------------------
Processing tree 0
Final rules from tree 0:
Rule 18:
     PL3 <= 1.7691
     QL2 <= 2.6701
     QL3 <= 1.7916
->  class Stable  [99.6%]
Rule 1:
     PL3 <= 1.8876
     QL2 <= 3.1796
     QL3 <= 1.3591
->  class Stable  [99.6%]
Rule 20:
     VG1 > 0.94
     PL3 <= 1.7691
     QL2 <= 2.6701
     QL3 <= 1.9895
->  class Stable  [99.5%]
...
<deleted>
...

Rule 34:
     PL2 > 3.1892
     QL2 > 2.6701
     QL3 > 1.3591
->  class UnStable  [95.2%]
```

```
Default class: Stable

Evaluation on training data (2000 items):
Rule  Size  Error  Used  Wrong           Advantage
----  ----  -----  ----  -----          ---------
  18     3   0.4%   344      0 (0.0%)      0 (0|0)  Stable
   1     3   0.4%    77      0 (0.0%)      0 (0|0)  Stable
  20     4   0.5%    47      0 (0.0%)      0 (0|0)  Stable
  59     4   0.5%    97      0 (0.0%)      0 (0|0)  Stable
  31     4   0.6%    53      0 (0.0%)      0 (0|0)  Stable
  ... <deleted>
  83     3   2.6%    13      1 (7.7%)      0 (0|0)  Stable
  44     2   2.8%    20      4 (20.0%)      0 (2|2)  Stable
  ... <deleted>
  43     4   4.1%     8      1 (12.5%)      6 (7|1)  UnStable
  34     3   4.8%     3      0 (0.0%)     3 (3|0)  UnStable

Tested 2000, errors 65 (3.2%)   <<

  (a)  (b) <-classified as
 ---- ----
  986   14 (a): class Stable
   52  948 (b): class UnStable


Evaluation on test data (4165 items):
Rule  Size  Error  Used  Wrong           Advantage
----  ----  -----  ----  -----          ---------
  18     3   0.4%   820      2 (0.2%)      0 (0|0)  Stable
  ... <deleted>
  34     3   4.8%     5      1 (20.0%)     3 (4|1)  UnStable

Tested 4165, errors 208 (5.0%)   <<
  (a)  (b) <-classified as
```

```
---- ----
2333   80 (a): class Stable
 128 1624 (b): class UnStable
-----------------------------------------------------
```

Figure 6.35: Classification by Production Rules (Case DT1)

After a similar preamble, the program generates rules from the decision tree of Figure 6.35. The first of them is rule number 18. The rule number (here 18) is arbitrary, being derived from the order of leaves in the original tree, and just serves to identify the rule. The left-hand side contains three tests and a case that classifies that satisfies all three of them is classified as *Stable*, the class identified by the right-hand side of the rule. The program also predicts that this classification will be correct for 99.6% of unseen cases that satisfy this rule's left-hand side. After listing similar rules, the programs select *UnStable* as the default class for this domain.

The set of rules is now evaluated on the training cases from which the tree was constructed, with the performance of each rule displayed separately. The statistics for the first rule (Rule 18) are:

This rule, with three tests in its left-hand side and a predicted error rate of 0.4%, was used 344 times in classifying the training cases. All of the cases that satisfied the rule's left-hand side did in fact belong to class *Stable*, so there are no errors in this set.

The section headed **Advantage** shows what would happen if this rule was omitted from the list. Each entry of the form $a(b|c)$ indicates that, if the rules were omitted, b classes now classified correctly by this rule would be classified incorrectly, and c cases now misclassified by this rule would be classified correctly by the subsequent rules and the default; the net benefit of retaining the rule is thus, $a = b - c$.

Following the report on each rule there is a *summary* and a *confusion matrix* showing how the misclassification of the training cases occurs. The production rule classification model is now evaluated on the unseen test cases. The performance of each rule is summarized as above.

The above tests were done using the *information gain* criterion. Some results using the *gain ratio* criterion are shown below. A discussion and formulation of the *information gain* and *gain ratio* criteria are given in Section 3.3.5.

```
------------------------------------------------------------
C4.5 decision tree generator
----------------------------------------
    Options:
Gain criterion used
Trees evaluated on unseen cases
Read 2000 cases (8 attributes) from bus3.data
Decision Tree:

Evaluation on training data (2000 items):
 Before Pruning          After Pruning
----------------   ---------------------------
Size      Errors   Size       Errors   Estimate
 161   25( 1.2%)    135   36( 1.8%)    ( 6.3%)   <<

Evaluation on test data (4165 items):
 Before Pruning          After Pruning
----------------   ---------------------------
Size      Errors   Size       Errors   Estimate
 161  164( 3.9%)    135  181( 4.3%)    ( 6.3%)   <<

 (a)  (b) <-classified as
---- ----
2299  114 (a): class Stable
  67 1685 (b): class UnStable
---------------------------------------------------------
```

Comments

- Even for a small size system, the size of the decision tree can be large which makes it difficult to comprehend. Translation of a decision tree into production rules eases the task of understanding the *knowledge* hidden in the data.
- The error in classifying power system classes is less using the *gain ratio* criterion as compared to *information gain* criterion.

2. **Case DT2:** This case is for IEEE 14 bus system. This system has three transformers with taps, five generator buses and nine load buses. In this case, the power system is classified as *Stable* or *Unstable* depending whether the load flow converged or not. The input comprises of tap ratio of all transformers (3), all generator voltages (5), and real and reactive loads at all the buses (14). All the input values are continuous.

Results

- The IEEE 14 bus system is tested using the *information gain* criterion.

```
------------------------------------------------------------
C4.5 decision tree generator
----------------------------------------

    Options:
Trees evaluated on unseen cases
Read 2000 cases (36 attributes) from bus14.data
Decision Tree: <deleted>

Evaluation on training data (2000 items):
 Before Pruning          After Pruning
----------------  ---------------------------
Size      Errors  Size      Errors  Estimate
 303   39( 1.9%)   287   45( 2.2%)    (11.0%)   <<

Evaluation on test data (4001 items):
 Before Pruning          After Pruning
----------------  ---------------------------
Size      Errors  Size      Errors  Estimate
```

```
  303  461(11.5%)    287  461(11.5%)    (11.0%)   <<

   (a)  (b) <-classified as
  ---- ----
  2446  396 (a): class Stable
    65 1094 (b): class UnStable
```

- The IEEE 14 bus system is tested using the *gain ratio* criterion.

```
-------------------------------------------------------------
C4.5 decision tree generator
----------------------------------------
    Options:
Gain criterion used
Trees evaluated on unseen cases
Read 2000 cases (36 attributes) from bus14.data
Decision Tree: <deleted>

Evaluation on training data (2000 items):
         Before Pruning         After Pruning
        ----------------  ----------------------------
        Size      Errors  Size      Errors  Estimate
         263   25( 1.2%)   249   31( 1.6%)    ( 9.3%)   <<

Evaluation on test data (4001 items):
 Before Pruning         After Pruning
----------------  ----------------------------
Size      Errors  Size      Errors  Estimate
 263  393( 9.8%)   249  400(10.0%)    ( 9.3%)   <<

  (a)  (b) <-classified as
 ---- ----
 2492  350 (a): class Stable
   50 1109 (b): class UnStable
-------------------------------------------------------------
```

Comments

- The error in classifying power system classes is less using the *gain ratio* criterion as compared to *information gain* criterion.

3. **Case DT3:** This case is same as case DT2, except that all the input values are pre-processed into 5 groups.

Results

- The IEEE 14 bus system is tested using the *information gain* criterion.

```
------------------------------------------------------------
C4.5 decision tree generator
----------------------------------------
    Options:
Trees evaluated on unseen cases
Read 2000 cases (36 attributes) from bus14a.data
Decision Tree: <deleted>

Evaluation on training data (2000 items):
         Before Pruning           After Pruning
        ----------------  ---------------------------
        Size      Errors  Size      Errors   Estimate
         943   61( 3.0%)   289  240(12.0%)    (22.2%)   <<

Evaluation on test data (4001 items):
         Before Pruning           After Pruning
        ----------------  ---------------------------
        Size      Errors  Size      Errors   Estimate
         943  565(14.1%)   289  696(17.4%)    (22.2%)   <<

          (a)  (b)     <-classified as
         ---- ----
         2284  558     (a): class Stable
          138 1021     (b): class UnStable
------------------------------------------------------------
```

- The IEEE 14 bus system is tested using the *gain ratio* criterion.

```
------------------------------------------------------------
C4.5  decision tree generator
----------------------------------------
    Options:
Gain criterion used
Trees evaluated on unseen cases
        Read 2000 cases (36 attributes) from bus14a.data
Decision Tree: <deleted>

Evaluation on training data (2000 items):
        Before Pruning           After Pruning
        ----------------  ---------------------------
        Size      Errors  Size      Errors   Estimate
         907    71( 3.5%)  259  252(12.6%)    (22.4%)   <<

Evaluation on test data (4001 items):
        Before Pruning           After Pruning
        ----------------  ---------------------------
        Size      Errors  Size      Errors   Estimate
         907  548(13.7%)   259  695(17.4%)    (22.4%)   <<

         (a)  (b)     <-classified as
        ---- ----
        2289  553     (a): class Stable
         142 1017     (b): class UnStable
------------------------------------------------------------
```

Comments

- Results in this case (DT3) are poor as compared to case (DT2) for the same system. In case DT3, the inputs were pre-processed into 5 classes. Using these results, and similar experiments on other data sets (not reported in this thesis) it appears that pre-processing of the input into classes does not help in this classification problem.

4. **Case DT4:** This case is for IEEE 14 bus system. The input comprises tap ratio of all transformers (3), all generator voltages (5), and real and reactive loads at all the buses (14), real power generation at buses 2 to 14 (leaving out the slack bus), and reactive power generation at all load buses (buses 6 to 14). All the input values are continuous. The induced decision tree classifies the power system into *Stable* or *Unstable* mode depending upon the input pattern.

Results

- The IEEE 14 bus system is tested using the *information gain* criterion.

```
------------------------------------------------------------
C4.5 decision tree generator
----------------------------------------

    Options:
        Trees evaluated on unseen cases
Read 2000 cases (58 attributes) from bus14b.data
Decision Tree:  <deleted>

Evaluation on training data (2000 items):
         Before Pruning          After Pruning
        ----------------   ---------------------------
        Size      Errors   Size      Errors   Estimate
         305  41( 2.0%)     289  47( 2.4%)    (11.2%)   <<

Evaluation on test data (4001 items):
         Before Pruning          After Pruning
        ----------------   ---------------------------
        Size      Errors   Size      Errors   Estimate
         305  474(11.8%)    289  474(11.8%)   (11.2%)   <<

          (a)  (b)      <-classified as
         ---- ----
         2435  407      (a): class Stable
           67 1092      (b): class UnStable
------------------------------------------------------------
```

- The IEEE 14 bus system is tested using the *gain ratio* criterion.

```
------------------------------------------------------------
C4.5  decision tree generator
----------------------------------------
    Options:
        Gain criterion used
        Trees evaluated on unseen cases
Read 2000 cases (58 attributes) from bus14b.data
Decision Tree:  <deleted>

Evaluation on training data (2000 items):
         Before Pruning           After Pruning
        ----------------   ---------------------------
        Size      Errors   Size      Errors   Estimate
         267   22( 1.1%)    253   28( 1.4%)    ( 9.2%)   <<

Evaluation on test data (4001 items):
         Before Pruning           After Pruning
        ----------------   ---------------------------
        Size      Errors   Size      Errors   Estimate
         267  386( 9.6%)    253  393( 9.8%)    ( 9.2%)   <<

          (a)  (b)      <-classified as
         ---- ----
         2497  345      (a): class Stable
           48 1111      (b): class UnStable
------------------------------------------------------------
```

Comments

- The error in this case (DT4) when compared with case (DT2), for the same system shows that adding more inputs (attributes) does not always help. In case DT4, 58 parameters were used as input as compared to 36 parameters in case DT2. Yet, the results are similar.

5. **Case DT5:** This case is for a 6 bus system. This system, has two transformers with taps, two generator buses and four load buses. The input consists of tap ratio of all transformers (2), all generator voltages (2), and real and reactive loads at all the buses (6). All the input values are continuous. The induced decision tree classifies the power system into *Stable* or *Unstable* mode depending upon the input pattern.

Results

- A 6 bus system is tested using the *information gain* criterion.

```
-------------------------------------------------------------
C4.5 decision tree generator
----------------------------------------
    Options:
        Trees evaluated on unseen cases
Read 2000 cases (16 attributes) from bus6.data
Decision Tree:  <deleted>

Evaluation on training data (2000 items):
         Before Pruning          After Pruning
         ----------------   ---------------------------
         Size      Errors   Size      Errors   Estimate
          305   36( 1.8%)    273   48( 2.4%)    (10.9%)   <<

Evaluation on test data (5288 items):
         Before Pruning          After Pruning
         ----------------   ---------------------------
         Size      Errors   Size      Errors   Estimate
          305  687(13.0%)    273  692(13.1%)    (10.9%)   <<

          (a)  (b)      <-classified as
         ---- ----
         3421  623      (a): class Stable
           69 1175      (b): class UnStable
-------------------------------------------------------------
```

- A 6 bus system is tested using the *gain ratio* criterion.

```
-------------------------------------------------------------
C4.5 decision tree generator
----------------------------------------
    Options:
        Gain criterion used
        Trees evaluated on unseen cases
Read 2000 cases (16 attributes) from bus6.data
Decision Tree:  <deleted>

Evaluation on training data (2000 items):
         Before Pruning          After Pruning
        ----------------   ---------------------------
        Size      Errors   Size      Errors   Estimate
         305   39( 1.9%)    271   51( 2.5%)    (11.0%)   <<

Evaluation on test data (5288 items):

         Before Pruning          After Pruning
        ----------------   ---------------------------
        Size      Errors   Size      Errors   Estimate
         305  620(11.7%)    271  634(12.0%)    (11.0%)   <<

         (a)  (b)      <-classified as
        ---- ----
        3476  568      (a): class Stable
          66 1178      (b): class UnStable
-------------------------------------------------------------
```

Comments

- The errors are less using the *gain ratio* criterion as compared to *information gain* criterion.
- The size of the system does not determine the rate of error. Cases (DT1). (DT2)

and (DT5) which are for 3, 14 and 6 bus systems respectively show same order of errors.

6. **Case DT6:** This case is for 6 bus system. All the input values are continuous. The input consists of tap ratio of all transformers (2), all generator voltages (2), and real and reactive loads at all the buses (6). The induced decision tree classifies the power system into *Stable*, *IntStable* (intermediate stable) or *Unstable* mode depending upon the input pattern. The class *IntStable* was classified by examining the load bus voltages and the corresponding **V**oltage **Q** (reactive power) **S**ensitivity (VQS) values. For VQS values above a certain limit (with load flow converged) the power system operating mode was classified as *IntStable*.

Results

- A 6 bus system is tested using the *information gain* criterion.

```
--------------------------------------------------------------
C4.5 decision tree generator
-----------------------------------------

    Options:
        Trees evaluated on unseen cases
Read 3000 cases (16 attributes) from bus6a.data
Decision Tree:  <deleted>

Evaluation on training data (3000 items):
         Before Pruning          After Pruning
         ----------------   ---------------------------
         Size      Errors   Size      Errors   Estimate
          243  37( 1.2%)     167  63( 2.1%)    ( 5.8%)   <<

Evaluation on test data (4288 items):
         Before Pruning          After Pruning
         ----------------   ---------------------------
         Size      Errors   Size      Errors   Estimate
          243  185( 4.3%)    167  192( 4.5%)    ( 5.8%)   <<
```

```
  (a)  (b)  (c) <-classified as
 ---- ---- ----
  984        41 (a): class Stable
      1244      (b): class UnStable
  151      1868 (c): class IntStable
--------------------------------------------------------------
```

- A 6 bus system is tested using the *gain ratio* criterion.

```
--------------------------------------------------------------
C4.5  decision tree generator
----------------------------------------

    Options:
        Gain criterion used
        Trees evaluated on unseen cases
Read 3000 cases (16 attributes) from bus6a.data
Decision Tree:  <deleted>
Evaluation on training data (3000 items):
         Before Pruning           After Pruning
        ----------------   ---------------------------
        Size      Errors   Size      Errors   Estimate
         219   29( 1.0%)    193   41( 1.4%)    ( 5.5%)   <<
Evaluation on test data (4288 items):
         Before Pruning           After Pruning
        ----------------   ---------------------------
        Size      Errors   Size      Errors   Estimate
         219  160( 3.7%)    193  172( 4.0%)    ( 5.5%)   <<
          (a)  (b)  (c) <-classified as
         ---- ---- ----
         1003        22 (a): class Stable
              1244      (b): class UnStable
          150      1869 (c): class IntStable
--------------------------------------------------------------
```

Comments

- **This case, presents an original way, developed in this thesis, of application of decision trees and the VQS Method of determining the voltage stability. While, the classification of a power system into *Stable* or *UnStable* states is useful [Hatziarqyriou94, Jeysurya90, Wehenkel95, Wehenkel93], the addition of another state, the *IntStable* state (standing for Intermediate Stable) can help in taking appropriate actions as the system moves into this state.**
- This case has the least error among all the cases studied above.
- Errors in classifying power system classes is less using the *gain ratio* criterion as compared to *information gain* criterion.

6.4.1 Summary on Decision Tree Induction

In the previous section, the performance of the algorithm for induction of decision tree for classifying power system operation mode was evaluated for a three bus, six bus, and IEEE 14 bus systems. The cases were:

1. A 3 bus system with all input parameters being continuous.
2. IEEE 14 bus system with all input parameters being continuous.
3. IEEE 14 bus system with all input parameters discretized into 5 classes.
4. Same as case 2, but with more input parameters. All parameters are continuous.
5. 6 bus system with all input parameters being continuous. The classification modes are *Stable* or *Unstable*.
6. 6 bus system with all input parameters being continuous. The classification modes are *Stable*, *IntStable* (standing for intermediate stable) or *Unstable*.

Table 6.30 summarizes the results:

Case No.	Trg. Data Info. Gain	Test Data Info. Gain	Trg. Data Gain Ratio	Test Data Gain Ratio
Case 1:	1.9%	4.9%	1.8%	4.3%
Case 2:	2.2%	11.5%	1.6%	9.3%
Case 3:	12.0%	17.4%	12.6%	17.4%
Case 4:	2.4%	11.8%	1.4%	9.8%
Case 5:	2.4%	13.1%	2.5%	12.0%
Case 6:	2.1%	4.5%	1.4%	4.0%

Table 6.30: Summary of Data Mining Results

The main findings are:

1. **A novel way of using decision tree for classifying power system operation mode has been proposed and illustrated, in this thesis.**
2. Decision tree coupled with more powerful voltage stability prediction methods (instead of the VQSM, used here) is very likely to produce better results and has the potential for on-line application.
3. Use of *gain ratio* criterion gives more accurate results as compared to *information gain* criterion.
4. The size of the power system under study does not affect the rate of errors. However, as the size of the power system increases, the size of the decision tree also increases.

6.5 Miscellaneous

6.5.1 Mixed Integer Linear Programming

[Hsu92] describes an expert system that combines the operator's dispatching rules and the integer linear programming routine (ILP) for voltage control of the Taiwan Power Company. The ILP is formulated to cater to this problem and minimize the number of switching of capacitors/inductors and/or adjustments to transformer taps.

The program in [Hsu92] has the following limitations: Only switchable capacitors/reactors and adjustable transformer taps are used as control variables. In addition, only the effect of control variables on load bus voltages is evaluated. The ILP approach does not consider generator VAr limitations.

This thesis investigates the applicability of Integer Linear Programming (ILP)/ Mixed Integer Linear Programming (MILP) to remedy Voltage/VAr violations problem. The idea is to examine whether the Linear Programming (LP) method can be replaced by an ILP/MILP method. Results obtained by the LP method for discrete devices have to be rounded up which may lead to errors.

Case VVC1 is studied for solving voltage/VAr violations by the ILP method, that is step changes for all controllers are integer. This case is for a 6 bus system. The system is at full-load level and has an under-voltage violation at bus number 3 as seen from A(old) values in Table 6.2. Step length for all controllers is assumed to be 0.01 p.u.. Solving this problem by LP suggests that controllers 1 to 6 should be adjusted by -12.5, -20, -4.6554, 10, 0 and 0 steps respectively. Controllers 1, 2 are tap changing transformers, 3, 4 are generator voltages, and 5, 6 are switchable capacitors (at bus numbers 3 and 6 respectively) as seen from Table 6.1.

Solving the above problem with ILP by the *branch and bound method* yielded unbounded results. Following this an attempt was made to find a MILP solution to the problem by forcing controller no. 1 to return an integer value. Still, the result was unbounded.

This simple example for a small power system illustrates that an ILP/MILP solution will not work for most of the problems. A better alternative seems is to use a linear programming method and using the results in conjunction with an expert system for deciding how to round the results appropriately.

6.5.2 Discrete and Eliminate Method

Reference [Grace93] reports a method called *discrete and eliminate* to solve an ILP problem. The method involves solving a LP problem. Following this, one non-integer variable is forced to be an integer (discretize) and the problem solved again (by keeping the value of the integer variables fixed, that is eliminate). This process is repeated until all variables are rounded to integers. There are two main problems with this method:

1. Iterative solution makes it very computation intensive.
2. Does not always produce results.

6.6 Conclusion to Chapter 6

This chapter has evaluated the performance of models and algorithms developed in this thesis and described in the previous chapter. The main findings are:

- A new model for computing the sensitivities associated with the tap changing transformer gave better results.

- It has been shown that the sensitivities can be used for load-shedding.

- A neural network based system is an enhancement over existing voltage/VAr correcting systems and load-sheddding schemes in the literature. Based on an off-line analysis, it computes new power flows over the transmission lines and transformers once the setting of controllers has changed.

- An improved power system state classifier by an induction decision tree based on data mining technology has been demonstrated.

- In this thesis, the time required for different methods was compared by using the throughput time to solve the same problem. A more accurate quantitative measure would be the CPU time required after the optimization of codes under comparison. This work has been identified as one of the future works.

Chapter 7

CONCLUSIONS

7.1 General

A fundamental motivation for knowledge-based software in power systems is the need to overcome the human cognitive barrier which the conventional Energy Management System (EMS) installations encounter during emergency operations. When sudden and unforeseen events occur in a power system, the operators have to understand the situation (diagnosis) and determine the actions (decisions) which will be required to return the system to a normal state. As all the tasks have to be done in real time, the operators are exposed to heavy mental stress and very often, their comprehension limits are reached. As the complexity of power system operations increases, without sufficient resources to cope with it, the human limitations are reached more easily and frequently.

In case of a disturbance, the operators need to understand what is happening on the power system. An artificial intelligence based program can provide guidance by presenting various scenarios that explain situations consistently. Operators can then check for the most plausible scenarios, some of which might have been overlooked. With this motivation in mind there has been considerable effort during recent years to develop expert systems for power system operation and planning [Madan97b][1]. This thesis addresses an operational problem of voltage/VAr violations and voltage stability. This research extends previous work and integrates the use of expert systems with other artificial intelligence techniques such as neural networks and data mining systems with analytical methods, for voltage and VAr problems in a novel manner.

---

[1] "Applications of Artificial Intelligence for Power Systems," *Electric Power Systems Research Journal*, vol. 41, no. 2, May 1997, pp. 117-131.

The state-of-the-art indicates that expert systems present a good solution to *ill-structured problems*, that is problems which are too difficult to handle by purely numerical algorithmic methods. This includes the problem of *voltage stability* and *voltage/VAr violation*. Artificial intelligence based problem solving and heuristic programming techniques can be effectively adapted for power system problems. It is one way of assisting human operators to evaluate the data and to advise them as to how to react to changing conditions.

7.2 Summary of Work Done

1. This research has investigated power systems operations in a systematic manner, with particular reference to voltage stability and voltage/VAr violation problems. Some of the key findings are:

    (a) Voltage/VAr violations problems are closely related to voltage stability problems because if voltage/VAr violations are left unattended they can develop into voltage instability problems.

    (b) A majority of recent large-scale system breakdowns have been the consequence of instabilities characterized by sudden voltage collapse phenomena [IEEE93]. The main reason for this has been the increase in the *transient stability* limits of power flows, permitting more power to be transferred over longer distances. This has been made possible by improved protection devices, improved voltage regulators, and the use of Static VAr Compensators (SVCs). Higher power flows and use of reactive power compensation have led to higher reactive power losses. This has made appropriate control of high voltage problematic in extreme situations, leading to voltage instabilities which have caused large scale blackouts.

    The investigation in this thesis served as a background for the design and development of an integrated hybrid intelligent strategy for combating voltage stability and voltage/VAr violation problems. This study is necessary in order to optimize the response to a power system emergency, caused by voltage stability and/or voltage/VAr violations.

2. A comprehensive analysis of available methods, technologies and current practices was done in this thesis. In particular, limitations in current practices in optimizing the response to a power system disturbance(s) were identified. The key findings are:

(a) The current practice in most utilities is to use the P-V and Q-V curves, which are obtained by repeated load flows and by artificially stressing the load buses one at a time. Their use is cumbersome and time consuming.

(b) Current load-shedding practices are based on load-shedding of a pre-determined customer if the need arises and use of under-voltage relays.

These limitations make current methods and methodologies unsatisfactory, and have motivated this research. The details are given in Chapter 2.

3. This project has made a critical study of applications of artificial intelligence and its sub-fields with a view to applying it to the enhancement of voltage stability and the remedying of voltage/VAr violations. Expert systems, neural networks, and data mining technologies have been used. The merits and demerits of each of these technologies has been examined and it has been shown how a combination of these technologies can overcome some of their individual weakness. Details are given in Chapters 3 and 4. This analysis of the application of AI technologies, in conjunction with the analytical methods, has led to the development of an integrated hybrid intelligent control strategy for voltage and VAr problems.

4. An integrated, hybrid, and intelligent system has been developed in this thesis that incorporates quality reasoning with analytical techniques. The developed system is a collection of programs called the **Comprehensive Voltage VAr Controller** (CVVC) toolbox which enables study of voltage stability and voltage/VAr control problems and their solutions.

   The CVVC toolbox has the following components:

   (a) A load flow module.

   (b) The **Voltage Q** (reactive power) **Sensitivity** (VQS) method that determines the voltage stability of the system based on the reduced Jacobian matrix.

   (c) The **Voltage VAr Controller Method** (VVCM) that finds suitable settings of controllers to rectify the voltage/VAr violations, based on an heuristic search. The VVCM incorporates an expert system based monitoring system and is called the **Expert System Monitor** (ESM). The ESM determines the presence of voltage/VAr violations and also indicates the severity of the problem.

   (d) The VVCM operation is based on the sensitivity of the controlled (dependent) variables w.r.t. the controller variables. The toolbox computes the sensitivity matrix, *[S]*. The sensitivity matrix describes the relationship between changes in the controller variables (tap setting of transformers, voltage of generator buses, and switching of capacitors/reactors at the load buses) and their effect on the dependent variables (reactive power output of generators and voltages at the load buses).

(e) The toolbox also incorporates methods which are supplementary to the VVCM and are based on *linear programming*, and *mixed integer linear programming* techniques for solving voltage/VAr violations.

(f) **Load Shedding Method(s)** that determine load-shedding location(s) and amount(s), when the system is voltage unstable or insufficient resources are available to correct severe voltage/VAr violations.

(g) The toolbox has algorithms for generating the *database* for training a neural network and for inducing a decision tree.

(h) An embedded neural network based system that predicts the new power flows over the transmission lines and transformers based on new controller settings as inputs.

(i) A decision-tree based classifier that classifies power system operating condition under *Stable*, *Intermediate-Stable* and *Unstable* conditions, based upon system parameters.

5. The developed product incorporates strategies that enable the adaptive, proactive, comprehensive, and fast decision making which is required in a complex viability crisis.

In short, this thesis documents how past experience (including simulation cases) can be stored in various forms such as production rules in the expert systems, a trained neural network, and a decision tree. On occurrence of an event (disturbance), this past experience can be applied to the solution of voltage and VAr problems. In addition, data mining algorithms help in the *acquisition of knowledge*, by inducing a decision tree from a set of examples.

7.2.1 Summary of Main Findings

1. The VVCM is the chosen method for correcting voltage/VAr violations.

2. The ILP/MILP routines do not always produce feasible results and are computationally more intensive than the LP routines.

3. The *Discrete and Eliminate* method is not suitable for the problem under consideration.

4. The method LSM-LP1 is the chosen method for calculating load-shedding.

5. A suitable database is essential for getting good results from a neural network based system and induced decision tree. This conclusion also follows from NN and DM applications to other domains.

6. The *gain ratio* criteria gives better results than the *information gain* criteria for inducing the decision tree.

7. The developed system can serve as a training tool for the operators to improve their skills for handling emergency conditions.

8. The problem under study may also be viewed as constrained resource allocation problem. Hence, some of the results may also be applied to areas such as message/packet routing in computer networks.

7.3 Contributions

This thesis has led to the development of a software package called the CVVC toolbox which can assist operators in making proper decisions when faced with *voltage stability* or *voltage/VAr violation* problems, in a comprehensive manner. The CVVC toolbox incorporates some key research ideas developed in this thesis. These include:

1. **New Transformer Model:** A new algorithm for calculating the sensitivity of the reactive power of generators and voltage of load buses w.r.t. a change in transformer tap ratio setting has been described (in Chapter 5) which gives more accurate results than other models reported in the literature.

2. **The VCCM:** The VCCM algorithm developed as a part of this research work has a number of improvements over other implementations reported in the literature. These are:

    (a) Implementation in a conventional language makes it faster.

    (b) Incorporation of weather/seasonal related information.

   (c) Ability to forecast new power flows over the transmission lines and transformers, once the settings of the controllers have been changed.

3. **Load-Shedding Methods:** Three models have been proposed in this thesis, which also have the ability to predict the effect of their actions on the rest of the system. The methods are:

   (a) **Load-shedding Method (LSM-SC)** based on the use of inverse of Sensitivity Coefficients(SC). Although the use of sensitivity coefficients for load-shedding, have been proposed in the literature, in this thesis, the sensitivity coefficients have been used in a novel way.

   (b) **Load-shedding Method (LSM-LP)** based on Linear Programming (LP) has been adapted from the literature which uses linear programming for voltage/VAr management.

   (c) **Load-shedding Method (LSM-DI)** based on computing the Δ Current (DI) injections required to ensure the desired voltage profile over the system, which has not been reported in the literature.

The developed CVVC toolbox presents a concise summary to the operator about the status of the power system and a list of corrective measures. The effect of these corrective measures on the rest of the system is also presented. The programs in the CVVC toolbox have the following features:

1. A hybrid system that combines the strengths of both analytical methods and AI techniques.

2. A system that can potentially improve upon the existing practice of using the V-Q, P-Q and S-V curves, by the utilities for voltage and VAr problems.

The advantages of the proposed scheme are:

1. **Adaptability:** The analysis of the test cases shows that the program is able to react properly to new situations which have not not been presented previously. The actions are adapted to the instantaneous power system state, because of the added intelligence.

2. **Proactive actions:** The system reacts when the system approaches a problem, and recommends actions before a more serious situation occurs. Unlike *under-voltage*

relay protection [Taylor92], there is no waiting for the *in-extremis* crisis. In addition, the voltage stability disturbances are properly classified because of the use of the induced decision tree algorithm as a pattern classifier, which has learned power system classification based on the data sets presented to it.

3. **Accuracy:** A comprehensive and accurate off-line analysis is incorporated in the developed system. The sensitivities are based on conditions existing before any voltage/VAr violations. The training of the neural network and the induction of decision tree are based on full load flow calculations.

4. **Coordination:** Before any action(s) are taken, the effect of these action(s) on the rest of the system are computed or predicted. If these actions(s) show that new violations would occur, the intended action(s) are modified. Hence all decisions are coordinated.

5. **Speed:** The speed of the developed system is one of the main advantages of the scheme. This is achieved by solving linear algebra equations for voltage/VAr violations and load-shedding, instead of nonlinear computations (such as in load-flow). Also, an *off-line* trained neural network is able to compute power flows based on new controller settings. In effect, these fast computations can quickly return the system to a more secure stable state. This can be followed by use of slower optimal corrective and restorative actions.

6. **Flexibility:** The developed system can be incrementally improved. The neural network can be retrained if the performance of the system during the testing period is not completely satisfactory. Similarly, the expert system can be made more capable by adding more rules and heuristics.

7. **Modularity:** The developed system is a modular system which can be incrementally enhanced by adding modules such as: predicting voltage stability, load forecasting, real and reactive power scheduling, etc.. This feature facilitates further enhancement of the developed product.

In summary, this research work makes a contribution to the state-of-the-art in voltage/VAr management, voltage stability, load-shedding, and intelligent engineering.

7.4 Future Research Directions

To be able to use the CVVC toolbox in a power system environment, the following steps for future investigation are required:

1. The studies performed in this thesis using the CVVC toolbox should be repeated on power system utility data and the results produced by this package should be compared with actual scenarios, in order to check the level of accuracy. Improvements can be made if necessary.

2. The CVVC toolbox should then be implemented in a *Dispatcher's Training Simulators* environment and its performance evaluated. Modifications can be made depending upon the users' comments.

3. The improved software from the previous step should be implemented in a power utility's EMS environment mode in the *off-line* mode and its performance critically evaluated.

4. Once enough confidence has been developed in the abilities of the developed software, it can be put on-line.

Some additional research directions are:

1. More research is needed for load-shedding method based on Δ current (DI) injections (LSM-DI) to realize its full potential.

2. A neural network based system has been developed for predicting power flows over the transmission lines and transformers. The input consists of a subset of control and dependent variables. The generated database can be potentially used for finding the setting of the controllers where the input is the desired values of the dependent variables.

3. Current power utilities operational manuals are full of *private knowledge* about the power system under their jurisdiction. This knowledge can be encoded in the next version of the developed expert system. The following is an example of a power utility's private knowledge:

   The G generator is the single most critical element in the network for maintaining the area security. It should be in service, especially during winter peaking months. Where *possible, all maintenance outages should be scheduled during the light load periods, in the months of spring and fall.*

4. Heuristics provide the means to prune search in expert systems and make them more powerful. Hence, the next logical step would be to continue to incorporate more heuristics in the programs by involving more human experts. Reference may be made to work of [Lachs90] where simple rules were derived for load-shedding based on simulation studies. An example is:

   Rule: The reactive power on any hydro or synchronous condenser unit should not increase by more than 30 MVAr, otherwise its terminal voltage will be raised to 106%.

   These kinds of rules make finding a solution much easier, as fewer computations are required. However, in this author's opinion, the accuracy of such methods has yet to be established and more research is needed. As noted earlier, the programs developed in this thesis can be extended in an incremental fashion.

5. Data mining research is presently focussed on *data cleaning* that is, to recognize insignificant data while subjecting it to data mining algorithms [Fayyad96]. This research when applied to power system data can significantly prune the decision tree. This makes understanding of data easier and can increase the usefulness of the discovered knowledge.

6. The performance of the data mining algorithm, used in this thesis (based on ID3), needs to be evaluated using the archived data of power system utilities. Further, cross-checking with different algorithms (such as CN2, DataExplorer)[2] could provide additional insights. It has been found empirically that when more algorithms are used the likelihood of accurate results is higher. The idea is that if one algorithm cannot find something, then another will.

7. More quantitative measures such as the CPU time requirements for different programs can be developed and comparison made with existing methods. This work will involve optimizing the codes under comparison which makes it a difficult task.

8. This thesis work involved the use of variety of languages and packages. Each of this package is well developed for a specific purpose. In this context, Integrated

---

[2] http://info.gte.com/

Distributed **I**ntelligent **S**ystem (IDIS) technology is a practical method to improve the capability of knowledge base management and maintenance [Rao91]. The IDIS is a large knowledge integration environment which integrates isolated expert systems, numerical programs, and commercial packages, by bringing them under the control of a *meta system*. Future work could involve exploiting the power of the IDIS environment. Typically a meta system involves interfacing with different types of data repositories. Some ideas for a meta system have also been explored in [Madan97f][3].

---

[3] "Artificial Intelligence Techniques for Mitigation of Interoperability of Distributed Heterogeneous Database Systems," *IEEE Canadian Conference on Electrical and Computer Engineering*, St. John's, Newfoundland, May 1997, pp. 245-248.

Appendix 'A'

CLIPS

CLIPS is a forward chaining rule-based language that has inferencing and representation capabilities similar to those of OPS5. Syntactically, CLIPS very closely resembles a subset of ART. ART, however, is a multi-paradigm tool which provides forward and backward chaining, hypothetical reasoning, a schema representation language, and object-oriented programming among other features. CLIPS provides only the forward chaining capabilities of ART. Backward chaining, hypothetical reasoning, a schema representation language and object-oriented programming paradigm are not provided in CLIPS.

CLIPS was designed at NASA/Johnson Space Center with the specific purposes of providing high portability, low cost, and easy integration with external systems. CLIPS was written using the C programming language to facilitate these objectives. CLIPS is an acronym for C Language Integated Production System. The C Language portion of the acronym can be somewhat misleading , since there is also an ADA version of CLIPS.

Because of its high portability, CLIPS has been installed on a wide variety of computers ranging from PCs to CRAY supercomputers.

Appendix 'B'

Summary of Programs Developed

The following is a summary of computer programs developed to test the algorithms developed in this thesis.

1. **Voltage reactive power (Q) Sensitivity Method (VQSM)**
    - **Purpose:** To find the voltage sensitivity of the load buses w.r.t. injection of reactive power. This enables determination of voltage stability/instability of the power system.
    - **Input:** All bus voltages, the topology of the connected network (admittance matrix).
    - **Environment:** Unix and MATLAB.
    - **Output:** The values of voltage sensitivity of the load buses w.r.t. the reactive power injection at the load buses.
    - **Comments:**
        - Adapted from method described in [Kundur94].
2. **Expert System Monitor (ESM)**
    - **Purpose:** To find the voltage/VAr violations and their severity and to make inferences based on existing power system data.
    - **Input:** All bus voltages, the topology of the connected network (admittance matrix), weather/seasonal information.
    - **Environment:** Unix, MATLAB, C, and CLIPS (an expert system shell).
    - **Output:** Identifies voltage/VAr violations and their severity.
    - **Comments:**
        - CLIPS is a forward chaining expert system shell.
3. **Voltage VAr Controller Method (VVCM)**
    - **Purpose:** To assist in remedying all voltage/VAr violations.
    - **Input:** The values of the control and dependent variables, the topology of the connected network (admittance matrix), and the lower and upper limits of the control and dependent variables.

- **Environment:** Unix and MATLAB
- **Output:** The recommended settings of the controllers and the new values of the dependent variables. The program also gives step by step calculations as voltage/VAr violations are rectified one by one.
- **Comments:** The developed system has following improvements over existing implementations:
  - Incorporates accurate model for calculating transformer sensitivities.
  - Incorporates weather/seasonal related information.
  - Implementation in a MATLAB environment allows easy *what-if* analysis and makes is faster as compared to implementation in AI languages, e.g. PROLOG.

4. **Load Shedding Methods (LSM)**

- **Purpose:** To recommend where and how much load to shed.
- **Input:** The values of the control and dependent variables, the topology of the connected network (admittance matrix), and the lower and upper limits of the control and dependent variables.
- **Environment:** Unix and MATLAB.
- **Output:** The recommended values and locations for load-shedding and the predicted values of the dependent variables.
- **Comments:** The developed systems have following improvements over existing methods:
  - The effect of load-shedding is predicted for the rest of the system.
  - The load-shedding method based on sensitivity coefficients (LSM-SC) is a new algorithm not reported in literature. This method is dependent upon the heuristics which are developed in this thesis.
  - The load-shedding method based on linear programming (LSP-LP1) is the chosen method for load-shedding and developed as a part of this thesis.

5. **Linear Programming Routines**

- **Purpose:** To solve voltage/VAr violation and load-shedding problems using linear programming routines.

- **Input:** The values of the control and dependent variables, the topology of the connected network (admittance matrix), and the lower and upper limits of the control and dependent variables.
- **Environment:** Unix and MATLAB.
- **Output:**
  - **For Voltage/VAr Violations:** New settings of the control variables and predicted values of the dependent variables.
  - **For Load-Shedding:** The amounts and location of loads to be shed, and predicted values of the dependent variables.
- **Comments:**
  - For voltage/VAr violation problem, the LP routines provide comparison with the VVCM solution.
  - For load-shedding problem, the LP routine is the chosen method for load-shedding.
  - LP routines for load-shedding have been developed as part of this research.

6. **Neural Network Based System - Mode 1 (Training)**

- **Purpose:** To train the ADALINE neural network.
- **Input (Training Data):** Tuples containing the input vectors as:

  $input = [(tap_1 ... tap_t); (V_i - V_j); (P_{gi} - P_{li}); (Q_{gi} - Q_{li}); (P_{gj} - P_{lj}); (Q_{gj} - Q_{lj})];$

  where

  tap_1 ... tap_t is the tap setting of t number of transformers

  $(V_i$ - $V_j)$ is the voltage difference between buses i and j

  $(P_{gi}$ - $P_{li})$ is the difference in real power generation and real power load at bus i

  $(Q_{gi}$ - $Q_{li})$ is the difference in reactive power generation and reactive power load at bus i

  $(P_{gj}$ - $P_{lj})$ is the difference in real power generation and real power load at bus j

  $(Q_{gj}$ - $Q_{lj})$ is the difference in reactive power generation and reactive power load at bus j.

  The output consists of the class (of power flow level) over the transmission line or transformer under consideration.
- **Environment:** Unix and MATLAB.

- **Output:** A trained neural network represented by values of the synaptic weights *[W]* and bias b.
- **Comments:**
  - The training is done in the off-line mode.
  - 2000 data sets are used for training the neural network based system.
  - This module has been developed as part of this research.

7. **Neural Network Based System- Mode 2 (Application)**

- **Purpose:** To use the trained ADALINE neural network to predict power flow level over the transmission lines and transformer once the settings of the controllers have been changed.
- **Input:** The input vector consists of:

  $input = [(tap_1 ... tap_t); (V_i - V_j); (P_{gi} - P_{li}); (Q_{gi} - Q_{li}); (P_{gj} - P_{lj}); (Q_{gj} - Q_{lj})];$

  where
  tap_1 ... tap_t is the tap setting of t number of transformers
  (V_i - V_j) is the voltage difference between buses i and j
  (P_{gi} - P_{li}) is the difference in real power generation and real power load at bus i
  (Q_{gi} - Q_{li}) is the difference in reactive power generation and reactive power load at bus i
  (P_{gj} - P_{lj}) is the difference in real power generation and real power load at bus j
  (Q_{gj} - Q_{lj}) is the difference in reactive power generation and reactive power load at bus j.

- **Environment:** Unix and MATLAB.
- **Output:** The power flow level over the transmission line or transformer connected between buses i and j.
- **Comments:**
  - This program enhances the VVCM and LSM programs by issuing a warning if in trying to correct a voltage/VAr related problem, power flow problems are introduced.
  - The limitations of a power system decoupled model are overcome.

    - This module has been developed as part of this research.

8. **Data Mining Based System - Mode 1 (Training)**
    - **Purpose:** To obtain an induction decision tree and set of production rules to predict the class of power system operation.
    - **Input (Training Data):** Tuples containing the input vectors as:
      All tap settings, all generator voltages, all real and reactive power loads. The output is one of the power system class; *Stable, Intermediate Stable or Unstable*. The decision regarding the above classification is made by using the VQS value from the VQSM program.
    - **Environment:** Unix, MATLAB and C (in the form of a program called the C4.5).
    - **Output:** An induced decision tree and a set of production rules.
    - **Comments:**
        - The training is done in the off-line mode.
        - 2000 data sets are used for training the data mining based system.
        - This module has been developed as a part of this research.

9. **Data Mining Based System - Mode 2 (Application)**
    - **Purpose:** To use the induced decision tree or set of production rules to classify the power system operating mode.
    - **Input:** The input vector consists of: all tap settings, all generator voltages, all real and reactive power loads.
    - **Environment:** Unix, MATLAB and C (in the form of a program called the C4.5).
    - **Output:** A decision whether the power system operating mode is *Stable, Intermediate Stable or Unstable* based on the power system input parameters.
    - **Comments:**
        - This module can replace the VQS method and speed up decision making regarding the class of power system operating mode.
        - This method introduces the concept of *Intermediate Stable* state which provides proactive capabilities.
        - This method has been developed as part of this research.

Appendix ‘C’

Data for Test Systems

Abbreviations

- G: Generator Bus
- L: Load Bus
- T: Transformer
- S: Slack Bus

Line Designation	Resistance p.u.	Reactance p.u.	Line Charging p.u.
1 - 6	0.123	0.518	0.0
1 - 4	0.080	0.370	0.0
4 - 6	0.097	0.407	0.0
5 - 6(T)	0.000	0.300	0.0
5 - 2	0.282	0.640	0.0
2 - 3	0.723	1.050	0.0
3 - 4(T)	0.000	0.133	0.0

B.1 Line Data for 6-Bus System

Bus Number (Type)	Starting Bus Voltage		Generation		Load	
	Magnitude p.u.	Phase Angle degrees	P_g p.u.	Q_g p.u.	P_l p.u.	Q_l p.u.
1 (S)	1.0	0	0.0	0.00	0.00	0.00
2 (G)	1.0	0	0.5	0.00	0.00	0.00
3 (L)	1.0	0	0.0	0.55	0.13	1.00
4 (L)	1.0	0	0.0	0.50	0.00	0.00
5 (L)	1.0	0	0.0	0.00	0.30	0.18
6 (L)	1.0	0	0.0	0.00	0.50	0.05

B.2 Operating Conditions for 6 Bus System

Bus Number	Minimum Volt. Magnitude p.u.	Maximum Volt. Magnitude p.u.	Minimum MVAr Capability p.u.	Maximin MVAr Capability p.u.
1	0.95	1.05	-3.00	3.00
2	0.95	1.05	-3.00	3.00

B.3 Regulated Bus Data for 6 Bus System

Transfomer Designation	Tap Settings	Minimum	Maximum	Step
5 - 6	1.00	0.8	1.2	0.01
3 - 4	1.00	0.8	1.2	0.01

B.4 Transformer Data for 6 Bus System

Line Designation	Resistance p.u.	Reactance p.u.	Line Charging p.u.
1 - 2	0.01938	0.05917	0.0264
1 - 7	0.05403	0.22304	0.0246
2 - 3	0.04699	0.19797	0.0219
2 - 6	0.05811	0.17632	0.0187
2 - 7	0.05695	0.17388	0.0170
3 - 6	0.06701	0.17103	0.0173
6 - 7	0.01335	0.04211	0.0064
6 - 8(T)	0.0	0.20912	1.0
6 - 9(T)	0.0	0.55618	1.0
4 - 7(T)	0.0	0.25202	1.0
4 - 11	0.09498	0.19890	0.0
4 - 12	0.12291	0.25581	0.0
4 - 13	0.06615	0.13027	0.0
8 - 5	0.0	0.17615	0.0
8 - 9	0.0	0.11001	0.0
9 - 10	0.03181	0.08450	0.0
9 - 14	0.12711	0.27038	0.0
10 - 11	0.08205	0.19207	0.0
12 - 13	0.22092	0.19988	0.0
13 - 14	0.17093	0.34802	0.0

B.5 Line Data for the IEEE 14 Bus System

Bus	Starting Bus Voltage		Generation		Load	
Number (Type)	Magnitude p.u.	Phase Angle degrees	P_g p.u.	Q_g p.u.	P_l p.u.	Q_l p.u.
1(S)	1.06	0.0	0.0	0	0.0	0.0
2(G)	1.022	0.0	0.4	0	0.217	0.127
3(G)	1.005	0.0	0.0	0	0.942	0.190
4(G)	1.0034	0.0	0.0	0	0.112	0.075
5(G)	1.044	0.0	0.0	0	0.0	0.0
6(L)	1.0	0.0	0.0	0	0.478	0.039
7(L)	1.0	0.0	0.0	0	0.076	0.018
8(L)	1.0	0.0	0.0	0	0.0	0.0
9(L)	1.0	0.0	0.0	0	0.295	0.166
10(L)	1.0	0.0	0.0	0	0.090	0.058
11(L)	1.0	0.0	0.0	0	0.035	0.018
12(L)	1.0	0.0	0.0	0	0.061	0.016
13(L)	1.0	0.0	0.0	0	0.135	0.058
14(L)	1.0	0.0	0.0	0	0.149	0.056

B.6 Operating Conditions for the IEEE 14 Bus System

Bus Number	Minimum Volt. Magnitude p.u.	Maximum Volt. Magnitude p.u.	Minimum MVAr Capability p.u.	Maximin MVAr Capability p.u.
1	0.95	1.10	-3.00	3.00
2	0.95	1.10	-4.00	5.00
3	0.95	1.10	0.00	4.00
4	0.95	1.10	-0.06	0.24
5	0.95	1.10	-0.06	0.24

B.7 Regulated Bus Data for IEEE 14 Bus System

Transfomer Designation	Tap Settings
6 - 8	1.00
6 - 9	1.00
4 - 7	1.00

B.8 Transformer Data for IEEE 14 Bus System

Line Designation	Resistance p.u.	Reactance p.u.	Line Charging p.u.
1 - 2	0.0192	0.0575	0.0264
1 - 7	0.0452	0.1852	0.0204
2 - 8	0.0570	0.1737	0.0184
7 - 8	0.0130	0.0379	0.0042
2 - 3	0.0472	0.1983	0.0209
2 - 9	0.0581	0.1763	0.0187
8 - 9	0.0119	0.0414	0.0045
3 - 10	0.0460	0.1160	0.0102
9 - 10	0.0267	0.0820	0.0085
9 - 11(T)	0.0120	0.0420	0.0045
9 - 12(T)	0.0	0.2080	0.978
9 - 10	0.0	0.5560	0.969
11 - 5	0.0	0.2080	0.0
11 - 12	0.0	0.1100	0.0
8 - 13(T)	0.0	0.2560	0.932
13 - 6	0.0	0.1400	0.0
13 - 14	0.1231	0.2559	0.0
13 - 15	0.0662	0.1304	0.0
13 - 16	0.0945	0.1987	0.0
14 - 15	0.2210	0.1997	0.0
16 - 17	0.0824	0.1923	0.0
15 - 18	0.1070	0.2185	0.0
18 - 19	0.0639	0.1292	0.0
19 - 20	0.0340	0.0680	0.0
12 - 20	0.0340	0.2090	0.0
12 - 17	0.0324	0.0845	0.0
12 - 21	0.0348	0.0749	0.0
12 - 22	0.0727	0.1499	0.0
21 - 22	0.0166	0.0236	0.0
15 - 23	0.1000	0.2020	0.0
22 - 24	0.1150	0.1790	0.0
23 - 24	0.1320	0.2700	0.0
24 - 25	0.1885	0.3200	0.0
25 - 26	0.2544	0.3800	0.0
25 - 27	0.1093	0.2087	0.0
28 - 27(T)	0.0,	0.3960	0.968
27 - 29	0.2198	0.4153	0.0
27 - 30	0.3202	0.6027	0.0
29 - 30	0.2399	0.4533	0.0
4 - 28	0.0636	0.2000	0.0214
9 - 28	0.0169	0.0599	0.0065

B.9 Line Data for IEEE 30 Bus System

Bus	Starting Bus Voltage		Generation		Load	
Number (Type)	Magnitude p.u.	Phase Angle degrees	P_g p.u.	Q_g p.u.	P_l p.u.	Q_l p.u.
1(S)	1.0	0	0.0	0.0	0.0	0.0
2(G)	1.0	0	0.4	0.0	0.217	0.127
3(G)	1.0	0	0.0	0.0	0.942	0.190
4(G)	1.0	0	0.0	0.0	0.300	0.300
5(G)	1.0	0	0.0	0.0	0.0	0.0
6(L)	1.0	0	0.0	0.0	0.0	0.0
7(L)	1.0	0	0.0	0.0	0.024	0.012
8(L)	1.0	0	0.0	0.0	0.076	0.016
9(L)	1.0	0	0.0	0.0	0.0	0.0
10(L)	1.0	0	0.0	0.0	0.228	0.109
11(L)	1.0	0	0.0	0.0	0.0	0.0
12(L)	1.0	0	0.0	0.0	0.058	0.02
13(L)	1.0	0	0.0	0.0	0.112	0.075
14(L)	1.0	0	0.0	0.0	0.062	0.01
15(L)	1.0	0	0.0	0.0	0.082	0.025
16(L)	1.0	0	0.0	0.0	0.035	0.018
17(L)	1.0	0	0.0	0.0	0.090	0.058
18(L)	1.0	0	0.0	0.0	0.032	0.009
19(L)	1.0	0	0.0	0.0	0.095	0.034
20(L)	1.0	0	0.0	0.0	0.022	0.007
21(L)	1.0	0	0.0	0.0	0.175	0.112
22(L)	1.0	0	0.0	0.0	0.0	0.0
23(L)	1.0	0	0.0	0.0	0.032	0.016
24(L)	1.0	0	0.0	0.0	0.087	0.067
25(L)	1.0	0	0.0	0.0	0.0	0.0
26(L)	1.0	0	0.0	0.0	0.035	0.023
27(L)	1.0	0	0.0	0.0	0.0	0.0
28(L)	1.0	0	0.0	0.0	0.0	0.0
29(L)	1.0	0	0.0	0.0	0.024	0.009
30(L)	1.0	0	0.0	0.0	0.106	0.019

B.10 Operating Conditions for IEEE 30 Bus System

Bus Number	Minimum Volt. Magnitude p.u.	Maximum Volt. Magnitude p.u.	Minimum MVAr Capability p.u.	Maximin MVAr Capability p.u.
1	0.95	1.05	-3.00	3.00
2	0.95	1.05	-3.00	3.00
3	0.95	1.05	-3.00	3.00
4	0.95	1.05	-3.00	3.00
5	0.95	1.05	-3.00	3.00
6	0.95	1.05	-3.00	3.00

B.11 Regulated Bus Data for IEEE 30 Bus System

Transfomer Designation	Tap Settings
9 - 11	0.978
9 - 12	0.969
8 - 13	0.932
28 - 27	0.968

B.12 Transformer Data for IEEE 30 Bus System

Appendix 'D'

Complete Decision Tree

```
-------------------------------------------------------------
C4.5 decision tree generator
----------------------------------------
    Options:
Trees evaluated on unseen cases
File stem <bus3>
Read 2000 cases (8 attributes) from bus3.data

Decision Tree:

QL3 <= 1.9895 :
|   PL3 <= 3.0859 :
|   |   QL3 <= 1.3591 :
|   |   |   QL2 <= 3.1796 :
|   |   |   |   PL3 <= 1.8876 : Stable (331.0)
|   |   |   |   PL3 > 1.8876 :
|   |   |   |   |   VG1 <= 0.93 :
|   |   |   |   |   |   QL2 > 2.7538 : UnStable (5.0)
|   |   |   |   |   |   QL2 <= 2.7538 :
|   |   |   |   |   |   |   QL3 <= 0.6761 : Stable (17.0)
|   |   |   |   |   |   |   QL3 > 0.6761 :
|   |   |   |   |   |   |   |   QL2 <= 0.8336 : Stable (8.0)
|   |   |   |   |   |   |   |   QL2 > 0.8336 :
|   |   |   |   |   |   |   |   |   PL2 > 2.1731 : UnStable (7.0)
|   |   |   |   |   |   |   |   |   PL2 <= 2.1731 :
|   |   |   |   |   |   |   |   |   |   PL3 <= 2.6636 : Stable (2.0)
|   |   |   |   |   |   |   |   |   |   PL3 > 2.6636 : UnStable (3.0/1.0)
|   |   |   |   |   VG1 > 0.93 :
|   |   |   |   |   |   QL2 <= 2.7104 : Stable (141.0/1.0)
|   |   |   |   |   |   QL2 > 2.7104 :
|   |   |   |   |   |   |   VG1 > 1 : Stable (16.0)
|   |   |   |   |   |   |   VG1 <= 1 :
|   |   |   |   |   |   |   |   QL3 <= 0.433 : Stable (5.0)
|   |   |   |   |   |   |   |   QL3 > 0.433 : UnStable (8.0/1.0)
|   |   |   QL2 > 3.1796 :
|   |   |   |   VG1 > 0.97 : Stable (23.0)
|   |   |   |   VG1 <= 0.97 :
|   |   |   |   |   PL3 <= 1.1238 : Stable (11.0)
|   |   |   |   |   PL3 > 1.1238 :
|   |   |   |   |   |   QL3 <= 0.3716 : Stable (3.0)
|   |   |   |   |   |   QL3 > 0.3716 :
|   |   |   |   |   |   |   QL3 > 0.6455 : UnStable (12.0)
|   |   |   |   |   |   |   QL3 <= 0.6455 :
|   |   |   |   |   |   |   |   TAP1 <= 0.95 : Stable (3.0/1.0)
|   |   |   |   |   |   |   |   TAP1 > 0.95 : UnStable (3.0)
|   |   QL3 > 1.3591 :
|   |   |   QL2 <= 2.6701 :
|   |   |   |   PL3 <= 1.7691 :
|   |   |   |   |   VG1 > 0.94 : Stable (112.0)
|   |   |   |   |   VG1 <= 0.94 :
|   |   |   |   |   |   QL3 <= 1.7916 : Stable (25.0)
|   |   |   |   |   |   QL3 > 1.7916 : UnStable (5.0/1.0)
|   |   |   |   PL3 > 1.7691 :
|   |   |   |   |   QL2 <= 0.962 :
|   |   |   |   |   |   VG1 > 0.92 : Stable (28.0)
|   |   |   |   |   |   VG1 <= 0.92 :
|   |   |   |   |   |   |   VG1 <= 0.9 : Stable (2.0)
|   |   |   |   |   |   |   VG1 > 0.9 : UnStable (2.0)
|   |   |   |   |   QL2 > 0.962 :
|   |   |   |   |   |   VG1 > 1.02 : Stable (10.0)
|   |   |   |   |   |   VG1 <= 1.02 :
|   |   |   |   |   |   |   PL2 <= 0.2923 : Stable (5.0/1.0)
|   |   |   |   |   |   |   PL2 > 0.2923 :
|   |   |   |   |   |   |   |   VG1 <= 0.96 : UnStable (37.0/1.0)
|   |   |   |   |   |   |   |   VG1 > 0.96 :
|   |   |   |   |   |   |   |   |   PL3 > 2.6045 : UnStable (6.0)
|   |   |   |   |   |   |   |   |   PL3 <= 2.6045 :
|   |   |   |   |   |   |   |   |   |   QL2 <= 2.336 : Stable (7.0)
|   |   |   |   |   |   |   |   |   |   QL2 > 2.336 : UnStable (3.0)
|   |   |   QL2 > 2.6701 :
|   |   |   |   VG1 > 1.04 : Stable (10.0)
|   |   |   |   VG1 <= 1.04 :
|   |   |   |   |   VG1 <= 0.97 : UnStable (44.0/1.0)
|   |   |   |   |   VG1 > 0.97 :
|   |   |   |   |   |   PL3 > 2.518 : UnStable (7.0)
|   |   |   |   |   |   PL3 <= 2.518 :
|   |   |   |   |   |   |   PL2 > 3.1892 : UnStable (3.0)
|   |   |   |   |   |   |   PL2 <= 3.1892 :
|   |   |   |   |   |   |   |   PL2 <= 2.3618 : Stable (10.0)
|   |   |   |   |   |   |   |   PL2 > 2.3618 :
|   |   |   |   |   |   |   |   |   PL2 <= 2.5913 : UnStable (2.0)
|   |   |   |   |   |   |   |   |   PL2 > 2.5913 : Stable (3.0)
|   PL3 > 3.0859 :
|   |   QL2 <= 1.3652 :
|   |   |   QL3 <= 1.0962 : Stable (32.0)
|   |   |   QL3 > 1.0962 :
|   |   |   |   VG1 <= 0.99 : UnStable (11.0/1.0)
|   |   |   |   VG1 > 0.99 : Stable (3.0)
|   |   QL2 > 1.3652 :
|   |   |   PL2 <= 0.1248 : Stable (5.0/1.0)
|   |   |   PL2 > 0.1248 :
|   |   |   |   VG1 <= 1.05 :
|   |   |   |   |   QL3 <= 0.1693 :
|   |   |   |   |   |   TAP1 <= 0.96 : Stable (3.0)
|   |   |   |   |   |   TAP1 > 0.96 : UnStable (2.0)
|   |   |   |   |   QL3 > 0.1693 :
|   |   |   |   |   |   QL3 <= 0.6761 :
|   |   |   |   |   |   |   VG1 <= 0.96 : UnStable (7.0)
|   |   |   |   |   |   |   VG1 > 0.96 : Stable (5.0/1.0)
|   |   |   |   |   |   QL3 > 0.6761 :
|   |   |   |   |   |   |   QL2 > 1.8074 : UnStable (44.0)
|   |   |   |   |   |   |   QL2 <= 1.8074 :
|   |   |   |   |   |   |   |   VG1 <= 1 : UnStable (11.0/1.0)
|   |   |   |   |   |   |   |   VG1 > 1 : Stable (3.0/1.0)
|   |   |   |   VG1 > 1.05 :
|   |   |   |   |   TAP1 <= 0.94 : UnStable (2.0)
|   |   |   |   |   TAP1 > 0.94 : Stable (8.0/1.0)
QL3 > 1.9895 :
|   QL2 <= 2.4881 :
|   |   QL3 <= 2.4983 :
|   |   |   PL3 <= 2.7314 :
|   |   |   |   VG1 <= 0.99 :
|   |   |   |   |   QL2 <= 1.0129 :
|   |   |   |   |   |   PL3 <= 1.5298 : Stable (15.0)
|   |   |   |   |   |   PL3 > 1.5298 :
|   |   |   |   |   |   |   VG1 <= 0.93 : UnStable (2.0)
|   |   |   |   |   |   |   VG1 > 0.93 : Stable (8.0/1.0)
|   |   |   |   |   QL2 > 1.0129 :
|   |   |   |   |   |   PL3 <= 0.6411 :
|   |   |   |   |   |   |   VG1 <= 0.95 : UnStable (3.0)
|   |   |   |   |   |   |   VG1 > 0.95 : Stable (5.0)
|   |   |   |   |   |   PL3 > 0.6411 :
|   |   |   |   |   |   |   QL2 > 1.3414 : UnStable (38.0)
|   |   |   |   |   |   |   QL2 <= 1.3414 :
|   |   |   |   |   |   |   |   PL3 <= 1.1499 : Stable (2.0)
|   |   |   |   |   |   |   |   PL3 > 1.1499 :
|   |   |   |   |   |   |   |   |   QL3 <= 2.131 : Stable (3.0/1.0)
|   |   |   |   |   |   |   |   |   QL3 > 2.131 : UnStable (6.0)
|   |   |   |   VG1 > 0.99 :
|   |   |   |   |   PL3 <= 2.3637 :
|   |   |   |   |   |   QL2 <= 2.336 : Stable (58.0)
|   |   |   |   |   |   QL2 > 2.336 :
|   |   |   |   |   |   |   QL2 <= 2.4053 : UnStable (3.0/1.0)
|   |   |   |   |   |   |   QL2 > 2.4053 : Stable (5.0)
|   |   |   |   |   PL3 > 2.3637 :
|   |   |   |   |   |   TAP1 > 1.03 : UnStable (3.0)
|   |   |   |   |   |   TAP1 <= 1.03 :
|   |   |   |   |   |   |   QL2 <= 2.3093 : Stable (7.0)
|   |   |   |   |   |   |   QL2 > 2.3093 : UnStable (2.0)
|   |   |   PL3 > 2.7314 :
|   |   |   |   QL2 > 0.4675 : UnStable (38.0)
|   |   |   |   QL2 <= 0.4675 :
|   |   |   |   |   VG1 <= 1.01 : UnStable (7.0/1.0)
|   |   |   |   |   VG1 > 1.01 : Stable (2.0)
|   |   QL3 > 2.4983 :
|   |   |   VG1 <= 1.01 :
|   |   |   |   PL3 <= 0.3142 :
|   |   |   |   |   QL2 > 1.2738 : UnStable (6.0)
|   |   |   |   |   QL2 <= 1.2738 :
|   |   |   |   |   |   VG1 <= 0.94 : UnStable (5.0/1.0)
|   |   |   |   |   |   VG1 > 0.94 : Stable (5.0)
|   |   |   |   PL3 > 0.3142 :
|   |   |   |   |   QL2 <= 0.1554 :
|   |   |   |   |   |   QL2 <= 0.0413 : Stable (2.0)
|   |   |   |   |   |   QL2 > 0.0413 :
|   |   |   |   |   |   |   VG1 <= 0.98 : UnStable (9.0)
|   |   |   |   |   |   |   VG1 > 0.98 :
```

```
|   |   |   |   |   |   |   |   QL3 <= 3.0213 : Stable (2.0)
|   |   |   |   |   |   |   |   QL3 > 3.0213 : UnStable (3.0)
|   |   |   |   |   QL2 > 0.1554 :
|   |   |   |   |   |   QL3 <= 2.5772 :
|   |   |   |   |   |   |   QL2 > 1.1083 : UnStable (14.0)
|   |   |   |   |   |   |   QL2 <= 1.1083 :
|   |   |   |   |   |   |   |   PL2 <= 2.0514 : Stable (3.0)
|   |   |   |   |   |   |   |   PL2 > 2.0514 : UnStable (4.0)
|   |   |   |   |   |   QL3 > 2.5772 :
|   |   |   |   |   |   |   PL2 > 0.5135 : UnStable (218.0)
|   |   |   |   |   |   |   PL2 <= 0.5135 :
|   |   |   |   |   |   |   |   PL3 > 1.119 : UnStable (37.0)
|   |   |   |   |   |   |   |   PL3 <= 1.119 :
|   |   |   |   |   |   |   |   |   PL2 <= 0.2732 : UnStable (7.0)
|   |   |   |   |   |   |   |   |   PL2 > 0.2732 :
|   |   |   |   |   |   |   |   |   |   VG1 <= 0.94 : UnStable (2.0)
|   |   |   |   |   |   |   |   |   |   VG1 > 0.94 : Stable (3.0/1.0)
|   |   |   VG1 > 1.01 :
|   |   |   |   PL3 <= 1.42 :
|   |   |   |   |   QL2 <= 1.3892 : Stable (27.0/1.0)
|   |   |   |   |   QL2 > 1.3892 :
|   |   |   |   |   |   PL2 <= 0.5762 : Stable (2.0)
|   |   |   |   |   |   PL2 > 0.5762 :
|   |   |   |   |   |   |   QL3 <= 2.6725 : Stable (2.0)
|   |   |   |   |   |   |   QL3 > 2.6725 : UnStable (13.0)
|   |   |   |   PL3 > 1.42 :
|   |   |   |   |   QL3 <= 2.7968 :
|   |   |   |   |   |   QL2 <= 0.5542 : Stable (6.0)
|   |   |   |   |   |   QL2 > 0.5542 :
|   |   |   |   |   |   |   QL3 > 2.7097 : Stable (2.0)
|   |   |   |   |   |   |   QL3 <= 2.7097 :
|   |   |   |   |   |   |   |   VG1 <= 1.05 : UnStable (10.0)
|   |   |   |   |   |   |   |   VG1 > 1.05 :
|   |   |   |   |   |   |   |   |   TAP1 <= 0.96 : Stable (2.0)
|   |   |   |   |   |   |   |   |   TAP1 > 0.96 : UnStable (3.0/1.0)
|   |   |   |   |   QL3 > 2.7968 :
|   |   |   |   |   |   QL2 > 0.6937 : UnStable (45.0)
|   |   |   |   |   |   QL2 <= 0.6937 :
|   |   |   |   |   |   |   PL3 > 2.5691 : UnStable (13.0)
|   |   |   |   |   |   |   PL3 <= 2.5691 :
|   |   |   |   |   |   |   |   TAP1 <= 0.95 : UnStable (2.0)
|   |   |   |   |   |   |   |   TAP1 > 0.95 : Stable (4.0)
|   QL2 > 2.4881 :
|   |   VG1 <= 1.06 : UnStable (266.0/1.0)
|   |   VG1 > 1.06 :
|   |   |   PL3 <= 0.7467 : Stable (2.0)
|   |   |   PL3 > 0.7467 : UnStable (19.0/1.0)

Simplified Decision Tree:

QL3 <= 1.9895 :
|   PL3 <= 3.0859 :
|   |   QL3 <= 1.3591 :
|   |   |   QL2 <= 3.1796 :
```

```
|   |   |   |   PL3 <= 1.8876 : Stable (331.0/1.4)
|   |   |   |   PL3 > 1.8876 :
|   |   |   |   |   VG1 <= 0.93 :
|   |   |   |   |   |   QL2 > 2.7538 : UnStable (5.0/1.2)
|   |   |   |   |   |   QL2 <= 2.7538 :
|   |   |   |   |   |   |   QL3 <= 0.6761 : Stable (17.0/1.3)
|   |   |   |   |   |   |   QL3 > 0.6761 :
|   |   |   |   |   |   |   |   QL2 <= 0.8336 : Stable (8.0/1.3)
|   |   |   |   |   |   |   |   QL2 > 0.8336 :
|   |   |   |   |   |   |   |   |   PL2 > 2.1731 : UnStable (7.0/1.3)
|   |   |   |   |   |   |   |   |   PL2 <= 2.1731 :
|   |   |   |   |   |   |   |   |   |   PL3 <= 2.6636 : Stable (2.0/1.0)
|   |   |   |   |   |   |   |   |   |   PL3 > 2.6636 : UnStable (3.0/2.1)
|   |   |   |   |   VG1 > 0.93 :
|   |   |   |   |   |   QL2 <= 2.7104 : Stable (141.0/2.6)
|   |   |   |   |   |   QL2 > 2.7104 :
|   |   |   |   |   |   |   VG1 > 1 : Stable (16.0/1.3)
|   |   |   |   |   |   |   VG1 <= 1 :
|   |   |   |   |   |   |   |   QL3 <= 0.433 : Stable (5.0/1.2)
|   |   |   |   |   |   |   |   QL3 > 0.433 : UnStable (8.0/2.4)
|   |   |   QL2 > 3.1796 :
|   |   |   |   VG1 > 0.97 : Stable (23.0/1.3)
|   |   |   |   VG1 <= 0.97 :
|   |   |   |   |   PL3 <= 1.1238 : Stable (11.0/1.3)
|   |   |   |   |   PL3 > 1.1238 :
|   |   |   |   |   |   QL3 <= 0.3716 : Stable (3.0/1.1)
|   |   |   |   |   |   QL3 > 0.3716 : UnStable (18.0/3.7)
|   |   QL3 > 1.3591 :
|   |   |   QL2 <= 2.6701 :
|   |   |   |   PL3 <= 1.7691 :
|   |   |   |   |   VG1 > 0.94 : Stable (112.0/1.4)
|   |   |   |   |   VG1 <= 0.94 :
|   |   |   |   |   |   QL3 <= 1.7916 : Stable (25.0/1.3)
|   |   |   |   |   |   QL3 > 1.7916 : UnStable (5.0/2.3)
|   |   |   |   PL3 > 1.7691 :
|   |   |   |   |   QL2 <= 0.962 :
|   |   |   |   |   |   VG1 > 0.92 : Stable (28.0/1.4)
|   |   |   |   |   |   VG1 <= 0.92 :
|   |   |   |   |   |   |   VG1 <= 0.9 : Stable (2.0/1.0)
|   |   |   |   |   |   |   VG1 > 0.9 : UnStable (2.0/1.0)
|   |   |   |   |   QL2 > 0.962 :
|   |   |   |   |   |   PL2 <= 0.2923 : Stable (8.0/2.4)
|   |   |   |   |   |   PL2 > 0.2923 :
|   |   |   |   |   |   |   VG1 <= 0.96 : UnStable (37.0/2.6)
|   |   |   |   |   |   |   VG1 > 0.96 :
|   |   |   |   |   |   |   |   PL3 > 2.6045 : UnStable (7.0/2.4)
|   |   |   |   |   |   |   |   PL3 <= 2.6045 :
|   |   |   |   |   |   |   |   |   QL2 <= 2.336 : Stable (13.0/1.3)
|   |   |   |   |   |   |   |   |   QL2 > 2.336 : UnStable (3.0/1.1)
|   |   |   QL2 > 2.6701 :
|   |   |   |   VG1 <= 0.97 : UnStable (44.0/2.6)
|   |   |   |   VG1 > 0.97 :
|   |   |   |   |   PL3 > 2.518 : UnStable (7.0/1.3)
|   |   |   |   |   PL3 <= 2.518 :
```

```
|   |   |   |   |   |   PL2 > 3.1892 : UnStable (3.0/1.1)
|   |   |   |   |   |   PL2 <= 3.1892 :
|   |   |   |   |   |   |   PL2 <= 2.3618 : Stable (14.0/1.3)
|   |   |   |   |   |   |   PL2 > 2.3618 :
|   |   |   |   |   |   |   |   PL2 <= 2.5913 : UnStable (2.0/1.0)
|   |   |   |   |   |   |   |   PL2 > 2.5913 : Stable (9.0/1.3)
|   PL3 > 3.0859 :
|   |   QL2 <= 1.3652 :
|   |   |   QL3 <= 1.0962 : Stable (32.0/1.4)
|   |   |   QL3 > 1.0962 :
|   |   |   |   VG1 <= 0.99 : UnStable (11.0/2.5)
|   |   |   |   VG1 > 0.99 : Stable (3.0/1.1)
|   |   QL2 > 1.3652 :
|   |   |   PL2 <= 0.1248 : Stable (5.0/2.3)
|   |   |   PL2 > 0.1248 :
|   |   |   |   VG1 <= 1.05 :
|   |   |   |   |   QL3 > 0.6761 : UnStable (58.0/5.0)
|   |   |   |   |   QL3 <= 0.6761 :
|   |   |   |   |   |   VG1 <= 0.96 : UnStable (10.0/2.4)
|   |   |   |   |   |   VG1 > 0.96 : Stable (7.0/2.4)
|   |   |   |   VG1 > 1.05 :
|   |   |   |   |   TAP1 <= 0.94 : UnStable (2.0/1.0)
|   |   |   |   |   TAP1 > 0.94 : Stable (8.0/2.4)
QL3 > 1.9895 :
|   QL2 > 2.4881 : UnStable (287.0/6.2)
|   QL2 <= 2.4881 :
|   |   QL3 <= 2.4983 :
|   |   |   PL3 <= 2.7314 :
|   |   |   |   VG1 <= 0.99 :
|   |   |   |   |   QL2 <= 1.0129 :
|   |   |   |   |   |   PL3 <= 1.5298 : Stable (15.0/1.3)
|   |   |   |   |   |   PL3 > 1.5298 :
|   |   |   |   |   |   |   VG1 <= 0.93 : UnStable (2.0/1.0)
|   |   |   |   |   |   |   VG1 > 0.93 : Stable (8.0/2.4)
|   |   |   |   |   QL2 > 1.0129 :
|   |   |   |   |   |   PL3 <= 0.6411 :
|   |   |   |   |   |   |   VG1 <= 0.95 : UnStable (3.0/1.1)
|   |   |   |   |   |   |   VG1 > 0.95 : Stable (5.0/1.2)
|   |   |   |   |   |   PL3 > 0.6411 :
|   |   |   |   |   |   |   QL2 > 1.3414 : UnStable (38.0/1.4)
|   |   |   |   |   |   |   QL2 <= 1.3414 :
|   |   |   |   |   |   |   |   PL3 <= 1.1499 : Stable (2.0/1.0)
|   |   |   |   |   |   |   |   PL3 > 1.1499 :
|   |   |   |   |   |   |   |   |   QL3 <= 2.131 : Stable (3.0/2.1)
|   |   |   |   |   |   |   |   |   QL3 > 2.131 : UnStable (6.0/1.2)
|   |   |   |   VG1 > 0.99 :
|   |   |   |   |   PL3 <= 2.3637 : Stable (66.0/3.8)
|   |   |   |   |   PL3 > 2.3637 :
|   |   |   |   |   |   TAP1 > 1.03 : UnStable (3.0/1.1)
|   |   |   |   |   |   TAP1 <= 1.03 :
|   |   |   |   |   |   |   QL2 <= 2.3093 : Stable (7.0/1.3)
|   |   |   |   |   |   |   QL2 > 2.3093 : UnStable (2.0/1.0)
|   |   |   PL3 > 2.7314 :
|   |   |   |   QL2 > 0.4675 : UnStable (38.0/1.4)
|   |   |   |   QL2 <= 0.4675 :
|   |   |   |   |   VG1 <= 1.01 : UnStable (7.0/2.4)
|   |   |   |   |   VG1 > 1.01 : Stable (2.0/1.0)
|   |   QL3 > 2.4983 :
|   |   |   VG1 <= 1.01 :
|   |   |   |   PL3 > 0.3142 : UnStable (304.0/11.8)
|   |   |   |   PL3 <= 0.3142 :
|   |   |   |   |   QL2 > 1.2738 : UnStable (6.0/1.2)
|   |   |   |   |   QL2 <= 1.2738 :
|   |   |   |   |   |   VG1 <= 0.94 : UnStable (5.0/2.3)
|   |   |   |   |   |   VG1 > 0.94 : Stable (5.0/1.2)
|   |   |   VG1 > 1.01 :
|   |   |   |   PL3 <= 1.42 :
|   |   |   |   |   QL2 <= 1.3892 : Stable (27.0/2.6)
|   |   |   |   |   QL2 > 1.3892 :
|   |   |   |   |   |   PL2 <= 0.5762 : Stable (2.0/1.0)
|   |   |   |   |   |   PL2 > 0.5762 :
|   |   |   |   |   |   |   QL3 <= 2.6725 : Stable (2.0/1.0)
|   |   |   |   |   |   |   QL3 > 2.6725 : UnStable (13.0/1.3)
|   |   |   |   PL3 > 1.42 :
|   |   |   |   |   QL3 <= 2.7968 :
|   |   |   |   |   |   QL2 <= 0.5542 : Stable (6.0/1.2)
|   |   |   |   |   |   QL2 > 0.5542 :
|   |   |   |   |   |   |   QL3 > 2.7097 : Stable (2.0/1.0)
|   |   |   |   |   |   |   QL3 <= 2.7097 :
|   |   |   |   |   |   |   |   VG1 <= 1.05 : UnStable (10.0/1.3)
|   |   |   |   |   |   |   |   VG1 > 1.05 :
|   |   |   |   |   |   |   |   |   TAP1 <= 0.96 : Stable (2.0/1.0)
|   |   |   |   |   |   |   |   |   TAP1 > 0.96 : UnStable (3.0/2.1)
|   |   |   |   |   QL3 > 2.7968 :
|   |   |   |   |   |   QL2 > 0.6937 : UnStable (45.0/1.4)
|   |   |   |   |   |   QL2 <= 0.6937 :
|   |   |   |   |   |   |   PL3 > 2.5691 : UnStable (13.0/1.3)
|   |   |   |   |   |   |   PL3 <= 2.5691 :
|   |   |   |   |   |   |   |   TAP1 <= 0.95 : UnStable (2.0/1.0)
|   |   |   |   |   |   |   |   TAP1 > 0.95 : Stable (4.0/1.2)

Tree saved

Evaluation on training data (2000 items):

 Before Pruning           After Pruning
----------------   ---------------------------
Size      Errors   Size      Errors   Estimate

 195   24( 1.2%)    149   39( 1.9%)    ( 6.8%)   <<

Evaluation on test data (4165 items):

 Before Pruning           After Pruning
----------------   ---------------------------
Size      Errors   Size      Errors   Estimate

 195  174( 4.2%)    149  205( 4.9%)    ( 6.8%)   <<
 (a)  (b) <-classified as
---- ----
2267  146 (a): class Stable
  59 1693 (b): class UnStable
------------------------------------------------------------------
```

Appendix 'E'

Complete Set of Production Rules

```
--------------------------------------------------------------
C4.5 rule generator
-------------------------------
    Options:
Rulesets evaluated on unseen cases
File stem <bus3>
Read 2000 cases (8 attributes) from bus3.data

Processing tree 0

Final rules from tree 0:

Rule 18:
     PL3 <= 1.7691
     QL2 <= 2.6701
     QL3 <= 1.7916
->  class Stable  [99.6%]

Rule 1:
     PL3 <= 1.8876
     QL2 <= 3.1796
     QL3 <= 1.3591
->  class Stable  [99.6%]

Rule 20:
     VG1 > 0.94
     PL3 <= 1.7691
     QL2 <= 2.6701
     QL3 <= 1.9895
->  class Stable  [99.5%]

Rule 59:
     VG1 > 0.99
     PL3 <= 2.3637
     QL2 <= 2.336
     QL3 <= 2.4983
->  class Stable  [99.5%]

Rule 31:
     VG1 > 0.97
     PL2 <= 2.3618
     PL3 <= 2.518
     QL3 <= 1.9895
->  class Stable  [99.4%]

Rule 12:
     PL3 <= 1.1238
     QL3 <= 1.3591
->  class Stable  [99.4%]
```

```
Rule 2:
     PL3 <= 3.0859
     QL2 <= 2.7538
     QL3 <= 0.6761
->  class Stable  [99.4%]

Rule 37:
     QL2 <= 1.3652
     QL3 <= 1.0962
->  class Stable  [99.4%]

Rule 11:
     VG1 > 1
     PL3 <= 3.0859
     QL3 <= 1.3591
->  class Stable  [99.4%]

Rule 23:
     VG1 > 0.92
     PL3 <= 3.0859
     QL2 <= 0.962
     QL3 <= 1.9895
->  class Stable  [99.3%]

Rule 8:
     VG1 > 0.93
     PL3 <= 3.0859
     QL2 <= 2.7104
     QL3 <= 1.3591
->  class Stable  [99.3%]

Rule 55:
     PL3 <= 1.1499
     QL2 <= 1.3414
     QL3 <= 2.4983
->  class Stable  [99.2%]

Rule 36:
     VG1 > 1.04
     PL3 <= 3.0859
     QL3 <= 1.9895
->  class Stable  [99.1%]

Rule 52:
     VG1 > 0.93
     PL3 <= 2.7314
     QL2 <= 1.0129
     QL3 <= 2.4983
->  class Stable  [98.8%]

Rule 54:
     VG1 > 0.95
     PL3 <= 0.6411
     QL2 <= 2.4881
     QL3 <= 2.4983
->  class Stable  [98.8%]

Rule 87:
     VG1 > 1.01
     QL2 <= 0.5542
     QL3 <= 2.7968
->  class Stable  [98.1%]

Rule 93:
     TAP1 > 0.95
     VG1 > 1.01
     PL3 <= 2.5691
     QL2 <= 0.6937
->  class Stable  [97.8%]

Rule 83:
     VG1 > 1.01
     PL3 <= 1.42
     QL2 <= 1.3892
->  class Stable  [97.4%]

Rule 44:
     VG1 > 0.96
     QL3 <= 0.6761
->  class Stable  [97.2%]

Rule 49:
     TAP1 > 0.94
     VG1 > 1.05
     QL3 <= 1.9895
->  class Stable  [97.1%]

Rule 69:
     VG1 > 0.94
     PL3 <= 0.3142
     QL2 <= 1.2738
->  class Stable  [97.0%]

Rule 89:
     TAP1 <= 0.96
     VG1 > 1.05
     QL2 <= 2.4881
     QL3 <= 2.7968
->  class Stable  [94.4%]

Rule 97:
     VG1 > 1.06
     PL3 <= 0.7467
->  class Stable  [89.4%]

Rule 58:
     VG1 <= 0.99
     PL3 > 0.6411
     QL2 > 1.3414
     QL3 > 1.9895
->  class UnStable  [99.5%]

Rule 53:
     VG1 <= 0.95
     QL2 > 1.0129
     QL3 > 1.9895
->  class UnStable  [99.5%]

Rule 86:
     PL2 > 0.5762
     QL2 > 1.3892
     QL3 > 2.6725
->  class UnStable  [99.4%]

Rule 95:
     PL3 > 1.42
     QL2 > 0.6937
     QL3 > 2.7968
->  class UnStable  [99.4%]

Rule 76:
     VG1 <= 1.01
     PL2 > 2.0514
     QL2 > 0.1554
     QL3 > 2.4983
->  class UnStable  [99.1%]

Rule 96:
     VG1 <= 1.06
     QL2 > 2.4881
     QL3 > 1.9895
->  class UnStable  [99.0%]

Rule 98:
     PL3 > 0.7467
     QL2 > 2.4881
     QL3 > 1.9895
->  class UnStable  [98.9%]

Rule 94:
     PL3 > 2.5691
     QL3 > 2.7968
->  class UnStable  [98.8%]

Rule 25:
     VG1 <= 0.96
     PL2 > 0.2923
     PL3 > 1.7691
     QL2 > 0.962
     QL3 > 1.3591
->  class UnStable  [98.6%]
```

```
Rule 35:
     PL3 > 2.518
     QL2 > 2.6701
     QL3 > 1.3591
->  class UnStable  [98.5%]

Rule 67:
     PL3 > 2.7314
     QL2 > 0.4675
     QL3 > 1.9895
->  class UnStable  [98.4%]

Rule 27:
     VG1 <= 1.02
     PL3 > 1.7691
     QL2 > 2.336
     QL3 > 1.3591
->  class UnStable  [98.4%]

Rule 47:
     VG1 <= 1.05
     PL2 > 0.1248
     PL3 > 3.0859
     QL2 > 1.8074
     QL3 > 0.6761
->  class UnStable  [98.3%]

Rule 30:
     VG1 <= 0.97
     QL2 > 2.6701
     QL3 > 1.3591
->  class UnStable  [98.3%]

Rule 78:
     VG1 <= 1.01
     PL3 > 0.3142
     QL3 > 2.5772
->  class UnStable  [97.8%]

Rule 6:
     VG1 <= 0.93
     PL2 > 2.1731
     PL3 > 1.8876
     QL2 > 0.8336
     QL3 > 0.6761
->  class UnStable  [97.5%]

Rule 65:
     VG1 <= 1.01
     PL3 > 2.7314
     QL3 > 1.9895
->  class UnStable  [97.3%]

Rule 10:
        VG1 <= 1
        PL3 > 1.8876
        QL2 > 2.7104
        QL3 > 0.433
   ->  class UnStable  [96.6%]

   Rule 38:
        VG1 <= 0.99
        PL3 > 3.0859
        QL3 > 1.0962
   ->  class UnStable  [96.2%]

   Rule 15:
        TAP1 > 0.95
        VG1 <= 0.97
        PL3 > 1.1238
        QL2 > 3.1796
        QL3 > 0.3716
   ->  class UnStable  [96.0%]

   Rule 43:
        VG1 <= 0.96
        PL3 > 3.0859
        QL2 > 1.3652
        QL3 > 0.1693
   ->  class UnStable  [95.9%]

   Rule 34:
        PL2 > 3.1892
        QL2 > 2.6701
        QL3 > 1.3591
   ->  class UnStable  [95.2%]

   Default class: Stable

   Evaluation on training data (2000 items):

   Rule  Size  Error  Used  Wrong            Advantage
   ----  ----  -----  ----  -----           ---------
     18     3   0.4%   344      0 (0.0%)       0 (0|0)   Stable
      1     3   0.4%    77      0 (0.0%)       0 (0|0)   Stable
     20     4   0.5%    47      0 (0.0%)       0 (0|0)   Stable
     59     4   0.5%    97      0 (0.0%)       0 (0|0)   Stable
     31     4   0.6%    53      0 (0.0%)       0 (0|0)   Stable
     12     2   0.6%    14      0 (0.0%)       0 (0|0)   Stable
      2     3   0.6%    70      0 (0.0%)       0 (0|0)   Stable
     37     2   0.6%    48      0 (0.0%)       0 (0|0)   Stable
     11     3   0.6%    37      0 (0.0%)       0 (0|0)   Stable
     23     4   0.7%    28      0 (0.0%)       0 (0|0)   Stable
      8     4   0.7%    12      1 (8.3%)       0 (0|0)   Stable
     55     3   0.8%    16      0 (0.0%)       0 (0|0)   Stable
     36     3   0.9%     7      0 (0.0%)       0 (0|0)   Stable
     52     4   1.2%    12      1 (8.3%)       0 (0|0)   Stable
     54     4   1.2%     5      0 (0.0%)       0 (0|0)   Stable
```

```
    87     3   1.9%    11      0 (0.0%)      0 (0|0)  Stable
    93     4   2.2%    15      0 (0.0%)      0 (0|0)  Stable
    83     3   2.6%    13      1 (7.7%)      0 (0|0)  Stable
    44     2   2.8%    20      4 (20.0%)     0 (2|2)  Stable
    49     3   2.9%     5      0 (0.0%)      0 (0|0)  Stable
    69     3   3.0%     5      0 (0.0%)      0 (0|0)  Stable
    89     4   5.6%     9      1 (11.1%)     0 (1|1)  Stable
    97     2  10.6%     4      1 (25.0%)    -1 (0|1)  Stable
    58     4   0.5%   299      0 (0.0%)     11 (11|0)  UnStable
    53     3   0.5%    59      0 (0.0%)      9 (9|0)  UnStable
    86     3   0.6%   116      0 (0.0%)     13 (13|0)  UnStable
    95     3   0.6%    63      0 (0.0%)     15 (15|0)  UnStable
    76     4   0.9%    46      0 (0.0%)      6 (6|0)  UnStable
    96     3   1.0%    39      1 (2.6%)     11 (12|1)  UnStable
    98     3   1.1%     9      1 (11.1%)     4 (5|1)  UnStable
    94     2   1.2%    26      0 (0.0%)     13 (13|0)  UnStable
    25     5   1.4%    70      1 (1.4%)     26 (27|1)  UnStable
    35     3   1.5%    20      0 (0.0%)      3 (3|0)  UnStable
    67     3   1.6%    30      0 (0.0%)      9 (9|0)  UnStable
    27     4   1.6%     8      1 (12.5%)     4 (5|1)  UnStable
    47     5   1.7%    25      0 (0.0%)      5 (5|0)  UnStable
    30     3   1.7%    24      1 (4.2%)     16 (17|1)  UnStable
    78     3   2.2%    67      6 (9.0%)     52 (57|5)  UnStable
     6     5   2.5%    11      0 (0.0%)      7 (7|0)  UnStable
    65     3   2.7%     7      1 (14.3%)     4 (4|0)  UnStable
    10     4   3.4%    17      0 (0.0%)     12 (12|0)  UnStable
    38     3   3.8%    13      1 (7.7%)      7 (8|1)  UnStable
    15     5   4.0%     3      0 (0.0%)      3 (3|0)  UnStable
    43     4   4.1%     8      1 (12.5%)     6 (7|1)  UnStable
    34     3   4.8%     3      0 (0.0%)      3 (3|0)  UnStable

Drop rule 97

Rule  Size  Error  Used  Wrong            Advantage
----  ----  -----  ----  -----           ---------
  18     3   0.4%   344      0 (0.0%)      0 (0|0)  Stable
   1     3   0.4%    77      0 (0.0%)      0 (0|0)  Stable
  20     4   0.5%    47      0 (0.0%)      0 (0|0)  Stable
  59     4   0.5%    97      0 (0.0%)      0 (0|0)  Stable
  31     4   0.6%    53      0 (0.0%)      0 (0|0)  Stable
  12     2   0.6%    14      0 (0.0%)      0 (0|0)  Stable
   2     3   0.6%    70      0 (0.0%)      0 (0|0)  Stable
  37     2   0.6%    48      0 (0.0%)      0 (0|0)  Stable
  11     3   0.6%    37      0 (0.0%)      0 (0|0)  Stable
  23     4   0.7%    28      0 (0.0%)      0 (0|0)  Stable
   8     4   0.7%    12      1 (8.3%)      0 (0|0)  Stable
  55     3   0.8%    16      0 (0.0%)      0 (0|0)  Stable
  36     3   0.9%     7      0 (0.0%)      0 (0|0)  Stable
  52     4   1.2%    12      1 (8.3%)      0 (0|0)  Stable
  54     4   1.2%     5      0 (0.0%)      0 (0|0)  Stable
  87     3   1.9%    11      0 (0.0%)      0 (0|0)  Stable
  93     4   2.2%    15      0 (0.0%)      0 (0|0)  Stable
  83     3   2.6%    13      1 (7.7%)      0 (0|0)  Stable
  44     2   2.8%    20      4 (20.0%)     0 (2|2)  Stable
```

```
   49     3    2.9%     5       0 (0.0%)       0 (0|0)  Stable
   69     3    3.0%     5       0 (0.0%)       0 (0|0)  Stable
   89     4    5.6%     9       1 (11.1%)       0 (1|1)  Stable
   58     4    0.5%   299       0 (0.0%)      11 (11|0)  UnStable
   53     3    0.5%    59       0 (0.0%)       9 (9|0)  UnStable
   86     3    0.6%   117       0 (0.0%)      14 (14|0)  UnStable
   95     3    0.6%    63       0 (0.0%)      15 (15|0)  UnStable
   76     4    0.9%    46       0 (0.0%)       6 (6|0)  UnStable
   96     3    1.0%    39       1 (2.6%)      11 (12|1)  UnStable
   98     3    1.1%     9       1 (11.1%)      4 (5|1)  UnStable
   94     2    1.2%    25       0 (0.0%)      13 (13|0)  UnStable
   25     5    1.4%    70       1 (1.4%)      26 (27|1)  UnStable
   35     3    1.5%    20       0 (0.0%)       3 (3|0)  UnStable
   67     3    1.6%    30       0 (0.0%)       9 (9|0)  UnStable
   27     4    1.6%     8       1 (12.5%)      4 (5|1)  UnStable
   47     5    1.7%    25       0 (0.0%)       5 (5|0)  UnStable
   30     3    1.7%    24       1 (4.2%)      16 (17|1)  UnStable
   78     3    2.2%    67       6 (9.0%)      52 (57|5)  UnStable
    6     5    2.5%    11       0 (0.0%)       7 (7|0)  UnStable
   65     3    2.7%     7       1 (14.3%)      4 (4|0)  UnStable
   10     4    3.4%    17       0 (0.0%)      12 (12|0)  UnStable
   38     3    3.8%    13       1 (7.7%)       7 (8|1)  UnStable
   15     5    4.0%     3       0 (0.0%)       3 (3|0)  UnStable
   43     4    4.1%     8       1 (12.5%)      6 (7|1)  UnStable
   34     3    4.8%     3       0 (0.0%)       3 (3|0)  UnStable

Tested 2000, errors 65 (3.2%)   <<

  (a)  (b) <-classified as
 ---- ----
  986   14 (a): class Stable
   52  948 (b): class UnStable

Evaluation on test data (4165 items):

Rule  Size  Error  Used  Wrong            Advantage
----  ----  -----  ----  -----           ---------
  18     3   0.4%   820      2 (0.2%)       0 (0|0)  Stable
   1     3   0.4%   185      1 (0.5%)       0 (0|0)  Stable
  20     4   0.5%    97      1 (1.0%)       0 (0|0)  Stable
  59     4   0.5%   220      4 (1.8%)       0 (0|0)  Stable
  31     4   0.6%   143      3 (2.1%)       4 (4|0)  Stable
  12     2   0.6%    28      1 (3.6%)       0 (0|0)  Stable
   2     3   0.6%   182      0 (0.0%)       0 (0|0)  Stable
  37     2   0.6%   112      0 (0.0%)       1 (1|0)  Stable
  11     3   0.6%    77      0 (0.0%)       0 (0|0)  Stable
  23     4   0.7%    61      1 (1.6%)       0 (0|0)  Stable
   8     4   0.7%    33      5 (15.2%)      0 (0|0)  Stable
  55     3   0.8%    41      1 (2.4%)      -1 (0|1)  Stable
  36     3   0.9%    19      1 (5.3%)       1 (1|0)  Stable
  52     4   1.2%    32      3 (9.4%)       0 (0|0)  Stable
  54     4   1.2%     9      0 (0.0%)       0 (0|0)  Stable
  87     3   1.9%    29      0 (0.0%)       0 (0|0)  Stable
  93     4   2.2%    31      1 (3.2%)       0 (0|0)  Stable
```

```
     83      3   2.6%    36        5 (13.9%)      0 (0|0)  Stable
     44      2   2.8%    44        7 (15.9%)      0 (5|5)  Stable
     49      3   2.9%    16        2 (12.5%)     -1 (0|1)  Stable
     69      3   3.0%    14        4 (28.6%)      1 (1|0)  Stable
     89      4   5.6%    24        4 (16.7%)     -2 (2|4)  Stable
     58      4   0.5%   499        6 (1.2%)      14 (18|4)  UnStable
     53      3   0.5%   105        0 (0.0%)      19 (19|0)  UnStable
     86      3   0.6%   229        5 (2.2%)      29 (34|5)  UnStable
     95      3   0.6%   110        1 (0.9%)      22 (23|1)  UnStable
     76      4   0.9%    86        2 (2.3%)      11 (12|1)  UnStable
     96      3   1.0%    85        6 (7.1%)      17 (20|3)  UnStable
     98      3   1.1%    16        4 (25.0%)      2 (6|4)  UnStable
     94      2   1.2%    36        0 (0.0%)      18 (18|0)  UnStable
     25      5   1.4%   112        6 (5.4%)      35 (41|6)  UnStable
     35      3   1.5%    34        0 (0.0%)       4 (4|0)  UnStable
     67      3   1.6%    58        5 (8.6%)      15 (20|5)  UnStable
     27      4   1.6%    15        2 (13.3%)      6 (8|2)  UnStable
     47      5   1.7%    51        1 (2.0%)       8 (9|1)  UnStable
     30      3   1.7%    35        3 (8.6%)      22 (24|2)  UnStable
     78      3   2.2%   120       19 (15.8%)     76 (94|18)  UnStable
      6      5   2.5%    20        2 (10.0%)     10 (12|2)  UnStable
     65      3   2.7%    11        2 (18.2%)      5 (5|0)  UnStable
     10      4   3.4%    34        6 (17.6%)     14 (20|6)  UnStable
     38      3   3.8%    25        4 (16.0%)      9 (13|4)  UnStable
     15      5   4.0%     4        0 (0.0%)       4 (4|0)  UnStable
     43      4   4.1%    14        5 (35.7%)      4 (9|5)  UnStable
     34      3   4.8%     5        1 (20.0%)      3 (4|1)  UnStable

Tested 4165, errors 208 (5.0%)   <<

  (a)  (b) <-classified as
 ---- ----
 2333   80 (a): class Stable
  128 1624 (b): class UnStable
------------------------------------------------------------------------
```

Bibliography

[Balu92] N.J. Balu, R. Adapa, G. Cauley, M. Lauby, and D.J. Maratukulam, "Review of Expert Systems in Bulk Power System Planning and Operation," *Proc. of the IEEE*, vol. 80, no. 5, May 1992.

[Barr82] A. Barr and E.A. Feigenbaum, (Eds.), *The Handbook of Artificial Intelligence*, vol. 2, Morgan Kaufmann Publishers, Los Altos CA, 1982.

[Barruncho92] L.M.F. Barruncho, J.P. Sucena Paiva, C.C. Liu, R. Pestana, and A. Vidigal, "Reactive Management and Voltage Monitoring and Control," *Electrical Power and Energy Systems*, vol. 14, no. 2/3, April/June 1992, pp. 144-157.

[Belhadj96] C. Belhadj, R. Mohamedi, S. Lefebvre, and X.-D. Do, "An Integrated Power System Global Controller Using Expert System," *Proc. Canadian Conference on Electrical and Computer Engineering*, May 26-29, 1996, Calgary, Alberta, pp. 446-449.

[Browell89] G. Browell and H. Clark, "Analysis and Solutions for Bulk System Voltage Instability," *IEEE Computer Applications in Power*, July 1989, pp. 31-35.

[Carbonell83] J.G. Carbonell, R.S. Michalski, and T.M. Mitchell, "An Overview of Machine Learning," *Machine Learning: An Artificial Intelligence Approach*, Morgan Kaufman Publishers Inc., 1983, pp. 1-23.

[Cheng88] S.J. Cheng, O.P. Malik, and G.S. Hope, "An Expert Systems for Voltage and Reactive Power Control of a Power System," *IEEE Trans. Power Systems*, vol. 3, no. 4, November 1988, pp. 1449-1455.

[Cheng89] S.J. Cheng, O.P. Malik, and G.S. Hope, "A Data Base Management System and its Application in a Voltage and Reactive Power Control Expert System," *Electric Power Systems Research*, vol. 16, no. 2, March 1989, pp. 147-155.

[Chang92] C.S. Chang and T.S. Chung, "An Expert System for Comprehensive On -Line Economic Security Controls of Distribution System," *Electric Power Systems Research*, vol. 24, 1992, pp. 165-172.

[Chowdhury93] B.H. Chowdhury, L. Swan, and D. Clark, "Expert System as a System Operator's Aid in Real-Time Solutions of the Optimal Power Flow," *Electric Power Systems Research*, vol. 26, no. 1, January 1993, pp. 21-30.

[Clips93] *CLIPS reference manual*, Lyndon B. Johnson Space Center, vols. I, II and III, 1993.

[Cote93] J.W. Cote and C.C. Liu, "Voltage Security Assessment Using Generalized Operational Planning Knowledge," *IEEE Trans. Power Systems*, vol. 8, no. 1, February 1993, pp.28-34.

[Doi85] A. Doi and K. Uemura, "Knowledge-Based Operation Guidance Method for Voltage and Reactive Power Control in Power Systems," *IFAC Electric Energy Systems*, 1985, pp.393-398.

[Dillon90] T.S. Dillon and M.A. Laughton (Eds.), *Expert Systems Applications to Power Systems*, U.K. Prentice Hall International, 1990.

[EPRI88] Voltage Stability and Security Assesment, EPRI EL-5967, August 1988.

[EPRI91] A Knowledge-Based Support System for Voltage Collapse Detection and Protection, EPRI EL-7168, February 1991.

[Fayyad96] U.M. Fayyad, G.P. Shapiro, P. Smyth, and R. Uthurusamy, *Advances in Knowledge Discovery and Data Mining*, AAAI Press/ The MIT Press, 1996.

[Frawley92] W.J. Frawley, G.P. Shapiro, and C.J. Mathews, "Knowledge Discovery in Databases: An overview," *Knowledge Discovery in Databases*, AAAI Press/ The MIT Press, 1991, pp. 1-27.

[Forgy83] C. Forgy, *OPS82 Users Manual*, Computer Science Department, Carnegie-Mellon University, 1983.

[Gao92] B. Gao, G.K. Morison, and P. Kundur, "Voltage Stability Evaluation Using Modal Analysis," *IEEE Tran. Power Systems*, vol. 7, no. 4, November 1992, pp. 1529-1542.

[Germond92] A.J. Germond and D. Niebur, "Survey of Knowledge-Based Systems in Power Systems: Europe," *Proc. of the IEEE*, vol. 80, no. 5, May 1992, pp. 732-744.

[Glover94] J.D. Glover and G.D. Digby, *Software Manual: Power System Analysis and Design*, PWS Publishing Company, Boston, 1994.

[Godart91] T.F. Godart and H.B. Puttgen, "A Reactive Path Concept Applied Within a Voltage Control Expert System," *IEEE Trans. Power Systems*, vol. 6, no. 2, May 1991, pp. 787-793.

[Goes92] N.L. Goes, E.F. Richards, and E.D. Tweed, "An Electric Control Center Operator's Assistant Expert System," *Electric Power Systems Research*, vol. 25, no. 3, December 1992, pp. 239-246.

[Grace93] A. Grace, *Optimization Toolbox for Use WithA Matlab*, The Math Work Inc., June 1993.

[Haykin94] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Maxwell Macmillan Canada, Inc., 1994.

[Hatziarqyriou94] N.D. Hatziarqyriou, G.C. Contaxis, and N.C. Sideris, "A Decision Tree Method for On-Line Steady State Security Assessment," *IEEE Trans. Power Systems*, vol. 9, no. 2, May 1994, pp. 1052-1061.

[Hobson80] E. Hobson, "Network Constrained Reactive Power Control Using Linear Programming," *IEEE Trans. Power Apparatus Systems*, vol. 99, no. 3, May/June 1980, pp. 868-877.

[Holsheimer94] M. Holsheimer and A. Seibes, *Data Mining: The Search for Knowledge in Databases*, Report CS-R9406, CWI, Amsterdam, 1994.

[Hong92] Y.Y. Hong, and C.C. Liu., "A Heuristic and Algorithmic Approach to VAr Planning," *IEEE Trans. Power Systems*, vol. 7, no. 2, May 1992, pp. 505-512.

[Hsu92] Y.Y. Hsu et al, "Voltage Control Using a Combined Integer Linear Programming and Rule-Based Approach," *IEEE Trans. Power Systems*, vol.7, no. 2, May 1992, pp. 744-752.

[Iba88] K. Iba, H. Suzuki, K. Suzuki, and K. Suzuki, "Practical Reactive Power Allocation/Operation Planning Using Successive Linear Programming," *IEEE Trans. Power Systems*, vol. 3, no. 2, May 1988, pp. 558-566.

[IEEE93] Voltage Stability of Power Systems: Concepts, Analytical Tools, and Industry Experience, IEEE Catalog Number 90TH0358-2, 1993.

[Jackson90] P. Jackson, *Introduction to Expert Systems*, Addison Wesley Publishing Company, 1990.

[Jeysurya90] B. Jeysurya and S.S. Venkata, "A Knowledge-Based Approach for Power System Dynamic Security Assessment," *ACM*, 6 1990, pp. 645-652.

[Kessel86] K.P. Kessel and H. Glavitsch, "Estimating the Voltage Stability of a Power System," *IEEE Trans. Power Delivery*, vol. 1, no. 3, July 1986, pp. 346-354.

[Keyhani89] A. Keyhani, S. Hao, and W.R. Wagner, "A Rule Based Approach to Construction of a Local Network Model for Decentralized Voltage Control," *Electric Power Systems Research*, vol. 16, 1989, pp. 119-126.

[Kono94] Y. Kono and A. Doi, "Development of a Knowledge-Based System for Voltage/Reactive Power Operation Planning," Electrical Engineering in Japan, vol. 111, no. 6, 1991, pp. 25-35.

[Kundur94] P. Kundur, *Power System Stability and Control*, McGraw-Hill, Inc., 1994.

[Lachs90] W. Lachs, Y.L. Zhu, D. Sutanto, and I.G. Morrison, "Automatic Control of System Voltage Stability by an Expert System," *Proc. of the Tenth Power System Computation Conference*, Graz, Austria, August 1990, pp. 1057-64.

[Lachs95] W.R. Lachs and D. Sutanto, "Rotor Heating as an Indicator of System Voltage Instability," *IEEE Trans. Power Systems*, vol. 10, no. 1, February 1995, pp. 175-181.

[Langley93] P. Langley, G.L. Bradshaw, and H.A. Simon, "Rediscovering Chemistry with the BACON System," *Machine Learning*, 1983, pp. 307-329.

[Le85] T.T. Le, M. Negnevistsky, and M. Piekutowski, "Expert System Application for Voltage Control and Var Compensation," *Engineering Intelligent Systems*, 1995, pp. 79-85.

[Liu90] C.C. Liu, H. Marathe, and K. Tomsovic "A Voltage Control Expert System and its Performance Evaluation," in *Expert Systems Applications to Power Systems*, Eds. T.S. Dillon and M.A. Laughton, U.K. Prentice Hall International, pp. 105-152.

[Madan96] S.Madan and K.E. Bollinger, "Artificial Intelligence Aided Voltage Stability Enhancement," *Canadian Conference on Electrical and Computer Engineering*, May 26th - May 29th, 1996, Calgary, Alberta, pp. 392-395.

[Madan97a] S. Madan and K.E. Bollinger, "Data Mining Applications for Power Systems," *VII th International Conference on Electronics, Communications and Computers, (CONIELECOMP'97)*, Puebla, Mexico, February 1997.

[Madan97b] S. Madan and K.E. Bollinger, "Applications of Artificial Intelligence for Power Systems," *Electric Power Systems Research Journal*, vol. 41, no. 2, May 1997, pp. 117-131.

[Madan97c] S. Madan, W.-K. Son, and K.E. Bollinger, "Applications of Data Mining for Power Systems," *Canadian Conference on Electrical and Computer Engineering*, St. John's, NF, May 1997, pp. 403-406.

[Madan97d] S. Madan, W.-K. Son, and K.E. Bollinger, "Development of an Integrated Intelligent System," *IEEE PACRIM'97*, Victoria, B.C., August 1997, pp. 827-830.

[Madan97e] S. Madan and K.E. Bollinger, "A Hybrid Intelligent System for Voltage and VAr Problems in Power Systems," *29th, North American Power Symposium*, Laramie, Wyoming, October 1997, pp. 76-83.

[Madan97f] S. Madan and R. Peters, "Artificial Intelligence Techniques for Mitigation of Interoperability of Distributed Heterogeneous Database Systems," *IEEE Canadian Conference on Electrical and Computer Engineering*, St. John's, Newfoundland, May 1997, pp. 245-248.

[Mamandur81] K.R.C. Mamandur and R.D. Chenoweth, "Optimal Control of Reactive Power Flow for Improvements in Voltage Profiles and for Real Power Loss Minimization," *IEEE Trans. Power Apparatus and Systems*, vol. 100, no. 7, July 1981, pp. 3185-3193.

[Mamandur82] K.R.C. Mamandur, "Emergency Adjustments to Var Control Variables to Alleviate Over-Voltages, Under-Voltages, and Generator VAR Limit Violations," *IEEE Trans. PAS*, vol. 101, no.5, May 1982, pp. 1040-1047.

[Matsuda88] S. Matsuda, H. Ogi, K. Nishimura, Y. Okataku, and S. Tamura, "Power System Voltage Control by Distributed Expert Systems," *International Workshop on Artificial Intelligence for Industrial Applications*, 1988, pp. 303-308.

[Michalski81] R.S. Michalski and R.L. Chilausky, "Knowledge Acquisition by Encoding Expert Rules Versus Computer Induction from Examples: a case study involving soybean pathology," in Mamdani and Gaines (eds), *Fuzzy Reasoning and Applications*, London Academic Press, 1981, pp. 247-271.

[Mota-Palomino86] R. Mota-Palomino and V.H. Quintana, "Sparse Reactive Power Scheduling by a Penalty Function - Linear Programming Technique," *IEEE Tran. Power Systems*, vol.1, no. 3, August 1986, pp. 31-39.

[Nilson90] N.J. Nilson, *Principles of Artificial Intelligence*, Springer International Student Edition, 1990.

[Palmer83] R.E. Palmer, R.C. Burchett, H.H. Happ, and D.R. Vierath, "Reactive Power Dispatching for Power System Voltage Security," *IEEE Trans. Power Systems*, vol. 102, no. 12, December 1983, pp. 3905-3909.

[Patterson96] D.W. Patterson, *Artificial Neural Networks: Theory and Application*, Prentice Hall, Singapore, 1996.

[Quinlan86] J.R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, 1986, pp.81-106.

[Quinlan93] J.R. Quinlan, *C4.5 Programs for Machine Learning*, Morgan Kaufmann Publishers, 1993.

[Rahman93] S. Rahman, "Artificial Intelligence in Electric Power Systems - A Survey of the Japanese Industry," *IEEE Trans. Power Systems*, vol. 8, no. 3, August 1993, pp. 1211-1218.

[Ramos95] J.L. Martinez Ramos et al, "A Hybrid Tool to Assist the Operator in Reactive Power/ Voltage Control and Optimization," *IEEE Trans. Power Systems*, vol.10, no. 2, May 1995, pp. 760-768.

[Rao91] M. Rao, *Integrated System for Intelligent Control*, Springer-Verlag, Berlin, 1991.

[Rao93] M. Rao, Q. Wang, and J. Cha, *Integrated Distributed Intelligent Systems in Manufacturing*, Chapman and Hall, 2-6 Boundary Row, London, 1993.

[Rekliatis83] G.V. Rekliatis, A. Ravindran, and K.M. Ragsdell, *Engineering Optimization:. Methods and Applications*, A Wiley-Interscience Publication, New York, 1983.

[Rich91] E. Rich and K. Knight, *Artificial Intelligence*, McGraw Hill, Inc., 1991.

[Sagakuchi88] T. Sagakuchi et al : "Prospects of Expert Systems in Power System Operation," *Electric Power and Energy Systems*, vol 10, no. 2, April 1988, pp. 71-82.

[Schafer91] K.F. Schafer, C. Schwartze, and J.F. Verstege, "CONTEX: an Expert System for Contingency Selection," *Electric Power Systems Research*, vol. 22, 1991, pp. 189-194.

[Simon83] H.A. Simon, "Why Should Machines Learn?," *Machine Learning: An Intelligence Approach*, Morgan Kaufmann Publishers, 1983, pp. 25-38.

[Schlueter87] R.A. Schlueter et al, "Voltage Stability and Security Assessment," EPRI Final Repor, PR-1999-8, Feb. 1987.

[Singh93] S.P. Singh, G.S. Raju, and A.K. Gupta, "Sensitivity Based Expert System for Voltage Control in Power System," *International Journal of Electrical Power and Energy System*, vol. 15, no. 3, June 1993, pp. 131-136.

[Stadlin82] W.O. Stadlin, and D.L. Fletcher, "Voltage Versus Reactive Current Model for Dispatch and Control," *IEEE Trans. Power Apparatus and Systems*, vol. 101, no. 10, October 1982, pp. 3751-3760.

[Stott74] B. Stott and O. Alsac, "Fast Decoupled Load Flow," *IEEE Trans. Power Systems*, vol. 93, no. 3, May 1974, pp. 859-869.

[Sun85] D.T. Sun and R.R. Shoults, "A Preventive Strategy Method for Voltage and Reactive Power Dispatch," *IEEE Trans. Power Apparatus and Systems*, vol. 104, no. 7, July 1985, pp. 1670-1676.

[Sutanto94] D. Sutanto, W.R. Lachs, and Y.L. Zhu, "Automatic Control of Voltage Stability by an Expert System Utilizing a Pattern Recognition Technique," *Engineering Intelligent Systems*, 1994, pp. 175-183.

[Synder84] W.L. Synder et al: "VAr Management - Problem Recognition and Control," *IEEE Trans. Power Apparatus and Systems*, vol. 103, no. 8, August 1984, pp. 2108-2116.

[Tamura83] H. Tamura, "Realstionship Between Voltage Instability and Multiple Load Flow Solutions in Electric Power Systems," *IEEE Trans. on Power Apparatus and Systems*, vol. PAS-402, no. 5, May 1983, pp. 1115-1125.

[Taylor94] C.W. Taylor, *Power System Voltage Stability*, McGraw-Hill, Inc., 1994.

[Taylor92] C.W. Taylor, "Concepts of Undervoltage Load Shedding for Voltage Stability," *IEEE Trans. Power Delivery*, vol. 7, no. 2, April 1992, pp. 480-488.

[Thomas86] R.J. Thomas et al, "On the Dynamics of Voltage Instabilities," *Proc. Bulk Power System Phenomena- Voltage Stability and Security*, EPRI EL-6183, PR 2473-21, January 1989.

[Vadari91] S.V. Vadari, *An Intelligent Load Shedding Approach to Enhance the Voltage Stability of a Power System*, Ph.D. Thesis, University of Washington, 1991.

[Vadari90] S.V. Vadari and S.S. Venkata, "Expert Load Shedding: a Hybrid Load Shedding Expert System in the Real-Time EMS Environment," Proc. of 10th PSCC, Graz, Austria, 19-24 August 1990, pp. 929-935.

[Vu88] K. Vu, Ta-Kang, Ma, R. Fischl, and C.C. Liu, "An Expert System for Voltage Security Assessment," *Symposium on Expert Systems Applications to Power Systems*, Stockholm-Helsinki, August 22-26, 1988, pp. 15-1 to 15-6.

[Wallach86] Y. Wallach, *Calculations and Programs for Power System Works*, Prentice Hall Inc., New Jersey, 1986.

[Wang90] S.M. Wang et al, "Inductive Learning in Power System Voltage Control." *Proc. of the 29th Conference on Decision and Control*, Honolulu, Hawaii, December 1990, pp. 3065-3066.

[Wagner90] W.R. Wagner, A. Keyhani, S. Hao, and T.C. Wong, "A Rule-Based Approach to Decentralized Voltage Control," *IEEE Trans. Power Ssytems*, vol. 5, no. 2, May 1990, pp. 643-651.

[Ward56] J.B. Ward and W.H. Hale, "Digital Computer Solution for Power-Flow Problems," *Trans. on AIEE*, vol. 75, III, 1956, pp. 398-404.

[Wehenkel93] L. Wehenkel and M. Pavella, "Decision Tree Approach to Power System Security Assessment, *Electrical Power and Energy Systems*, vol. 15, no. 1, February 1993, pp. 13-36.

[Wehenkel95] L. Wehenkel, *Machine Learning Approaches to Power System Security Assessment*, University of Liege, Belgium, 1995.

[Widrow85] B.D. Widrow and S.D. Stearns, *Adaptive Signal Processing*, Prentice Hall Inc., New Jersey, 1985.

[Wong93] K.P. Wong, "Artificial Intelligence and Neural Network Applications in Power Systems," *IEE 2nd International Conferences on Advances in Power System Control, Operation and Management*, Hong-Kong, December 1993, pp. 37-46.

[WorkingGroup88] *Expert System Applied to Voltage and Var Control*, CIGRE Working Group 38.06.01, pp. 109-131.

[Wu91] Y.H. Wu and S. Wang, "Discovering Functional Relationships from Observational Data," *Knowledge Discovery In Databases*, AAAI/ The MIT Press, 1991, pp. 55-92.

[Yang94] C.C. Yang and Y.Y. Hsu, "Estimation of Line Flow and Bus Voltages Using Decision Trees," *IEEE Trans. Power Systems*, vol. 9, no. 3, August 1994, pp. 1569-1574.

[Yokoyama93] R. Yokoyama, T. Niimura, and Y. Nakanishi, "Coordinated Control of Voltage and ReaCctive Power by Heuristic Modeling and Approximate Reasoning," *IEEE Trans. Power Systems*, vol. 8, no. 2, May 1993, pp. 636-645.

[Zaborszky79] J. Zaborszky, K.W. Whang, and K.W. Prasad, "Monitoring Evaluation and Control of Power System Emergencies," *Proceedings- System Engineering for Power Conference*, Enginnering Foundation Report CONF-790904-P1, Davos, Switzerland, October, 1979.

[Zalapa95] R.R. Zalapa and B.J. Cory, "Reactive Reserve Management," *IEE Proc.-Gener. Transm. Distrib.*, vol. 142, no. 1, January 1995.

[Zhang89] Z.Z. Zhang, G.S. Hope, and O.P, Malik, "Expert Systems in Electric Power Systems - A Bibliographical Survey," IEEE Trans. Power Systems, vol. 4, no. 4, October 1989, pp. 1355-1361.